

Document downloaded from:

<http://hdl.handle.net/10251/54606>

This paper must be cited as:

Hernandez, L.; Baladron, C.; Aguiar, JM.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. (2014). Artificial Neural Networks for Short-Term Load Forecasting in Microgrids Environment Energy. Energy. 75:252-264. doi:10.1016/j.energy.2014.07.065.



The final publication is available at

<http://dx.doi.org/10.1016/j.energy.2014.07.065>

Copyright Elsevier

Additional Information

Artificial Neural Networks for Short-Term Load Forecasting in Microgrids Environment

Luis Hernández¹, Carlos Baladrón², Javier M. Aguiar², Belén Carro², Antonio Sánchez-Esguevillas², Jaime Lloret³

¹*CIEMAT, Avenida Complutense 22, 28040 Madrid (Spain)*
+34 975 28 10 13
+34 975 28 10 51
luis.hernandez@ciemat.es

²*Dpto. TSyCeIT, ETSIT, Universidad de Valladolid, Paseo de Belén 15, 47011 Valladolid (Spain)*
+34 983 42 36 66
+34 983 42 36 67
C.B.: cbalzor@ribera.tel.uva.es; J.M.A.: javagu@tel.uva.es; B.C.: belcar@tel.uva.es; A.S.-E.: anisan@tel.uva.es

³*Departamento de Comunicaciones, Universidad Politécnica de Valencia, Camino Vera, s/n, 46022, Valencia (Spain)*
+34 609 54 90 43
jlloret@ocom.upv.es

Abstract

The adaptation of energy production to demand has been traditionally very important for utilities in order to optimize resource consumption. This is especially true also in *microgrids* where many intelligent elements have to adapt their behaviour depending on the future generation and consumption conditions. However, traditional forecasting has been performed only for extremely large areas, such as nations and regions. This work aims at presenting a solution for *Short-Term Load Forecasting (STLF)* in *microgrids*, based on a three-stage architecture which starts with pattern recognition by a *Self-Organizing Map (SOM)*, a clustering of the previous partition via *k-means* algorithm, and finally demand forecasting for each cluster with a *MultiLayer Perceptron*. Model validation was performed with data from a *microgrid*-sized environment provided by the Spanish company Iberdrola.

Keywords: Artificial Neural Network, Short-Term Load Forecasting, Microgrid, Pattern Recognition, Self-Organizing Map, k-means Algorithm, MultiLayer Perceptron.

1. Introduction

Electricity generation in the classical model of electric power systems is centralized in a very small number of elements, big power plants, which are normally located far away from final users. This model, while presenting many advantages for facilitating control and management of energy production, results in several problems such as transportation losses (caused by the long distances travelled by the energy) or problems, and long reaction times for adaptation of the network to dynamic requirements.

During the last years, national governments and international organizations have been developing and implementing strategies to achieve energy efficiency and low carbon emissions, and in general terms, to guarantee a more sustainable energy production system. Particularly in Europe, the *Strategic Energy Technology Plan (SET-Plan)* [1] established by the European Commission has been developed to coordinate, manage and foster efforts to this end across the continent. The European Industrial Initiative on the electricity grid aims at developing, demonstrating and validating, at scale, the technologies, system integration and processes to: enable the transmission and distribution of dispersed and concentrated renewable sources, integrate national networks into a truly pan-European network, optimise the investments and operational costs involved in upgrading the European electricity networks, guarantee a high quality of electricity supply to all customers and engage them as active participants in energy efficiency, and anticipate new developments such as the electrification of transport. In order to face all these challenges it is necessary to introduce enhanced intelligence and planning capabilities into the network, and therefore, to monitor and anticipate the behavior of demand and generation (renewable generation sources) using forecasting mechanisms, and specifically, the systems and algorithms studied along this paper.

One of the technological solutions devised to facilitate adaptation of generation and demand is the *Smart Grid (SG)*, an intelligent network based on Information and Communication Technologies to optimize the distribution grid. *SGs* are autonomic environments which not only can assist in the coordination of resources to accomplish collaborative tasks but are also

capable of self-management to reduce operators' interventions as much as possible. Tang *et al.* [2] present for instance an autonomic real-time transaction service that can dynamically discover grid services as participants to execute specified sub-transactions, coordinate these participants to achieve the real-time and transactional requirements, and assign priorities to schedule concurrent transactions.

According to the *Consortium for Electric Reliability Technology Solutions (CERTS)*, a *microgrid* is: "...an aggregation of loads and microsources operating as a single system providing both power and heat (...)" (Lasseter *et al.* [3]). A *microgrid* is conceived as a place to generate and consume energy more efficiently, and for that a *microgrid* needs support applications and services.

One of those required support services is demand forecasting. In the case of *microgrids*, these forecasts are essential, allowing the precise adjustment of generation and consumption. While the state of the art is rich in the study of demand forecasting in large environments (nations or regions), there is a lack of scientific literature regarding demand forecasting in *microgrids*. The aim of this work is therefore to develop a forecasting method suitable for these microgrid-sized environments and evaluate its results, including also a methodology to control error evolution depending on the amount of data available.

After this introduction, Section 2 presents the state of the art of models based on *ANN*, which are not applied in *microgrids*. Section 4 presents the proposed architecture and research data, and Section 3 details the models used in this architecture. Section 5 presents the results obtained, which show that the model described is optimal. Section 6 shows the distribution of errors, errors in a weekday and month, and the dependence of the error on the number of learning patterns. Finally, Section 7 summarizes the conclusions and future work.

2. Related Works

Utilities are always trying to adjust power production to the specific requirements of the demand they are covering, since the storage of electrical energy is expensive. This is also true for *microgrids*, where load curves are much more irregular and dynamic than national or

regional environments (due to the disaggregation of loads), and therefore more difficult to forecast. Some initial works regarding data processing in *microgrid* environments are being carried out. For instance, Venayagamoorthy and Welch [4] present two energy dispatch controllers for use in a grid-independent photovoltaic (*PV*) system; the solution of the energy dispatch controllers presented in this paper is based on an *ANN*. Hernández *et al.* [5] show a data processing system to cluster energy consumption patterns in industrial parks (potential *microgrids*).

There are different criteria to classify the demand forecasting models traditionally applied to large areas. With regard to the forecasting interval, different typologies can be identified: very short-term forecasting; short-term forecasting (*STLF*); and .3medium-term [6] and long-term forecasting [7]. Regarding the number of forecasting values, there are two main groups: a single value or more-than-one value.

Therefore, it is possible to forecast a value or multiple values. Concerning the first group, different techniques using *ANN* can be found in the literature; Park *et al.* [8] and Ho *et al.* [9] use *ANNs* for forecasting certain hours of the following day and peak load respectively; Ho *et al.* [10] and Dash [11] apply the forecasting of the peak load to feed an expert system and obtain a load curve; in Drezga and Rahman [12,13] and McMenemy and Monforte [14] multiple *ANNs* in parallel based on a single value are used.

Regarding the second group, that is forecasting using multiple values, some good examples can also be identified: Lee *et al.* [15] divide the day into three periods and make an independent forecasting; Lu *et al.* [16] use data from two utilities, adjusting each model separately; Papalexopoulos *et al.* [17] and Bakirtzis *et al.* [18] use nonlinear functions of temperature and setting of holidays; finally, Lou and Dong [19] present a real application based on type-2 (T2) fuzzy sets. Deihimi and Showkati [20] present the application of *Echo State Network (ESN)* to the *STLF* problem in power systems for 24-h ahead predictions while using the least number of inputs: current-hour load, predicted target-hour temperature, and only for 24-h ahead forecasting, day-type index.

Self-Organizing Maps (SOM) have also been used in different ways in the past; for example, Marín *et al.* [21] apply *SOM* for the classification of historical data; Joya *et al.* [22] use *SOM* for forecasting the load and analysis of contingencies; Mori and Itagaki [23] use *SOM* for the classification of data and combines it with *Radial Basis Function Network (RBFN)*; Carpinteiro and Reis [24] apply two *SOM* in cascade; Wang [25] uses *SOM* and fuzzy for forecasting; Fan and Chen [26] show two different stages in the forecasting process: a first stage with *SOM* is used to classify and in the second, 24 *Support Vector Machine (SVM)* are applied to every group. In Farhadi and Tafreshi [27], *SOM* is used to classify normal and abnormal days, and a *MultiLayer Perceptron (MLP)* to manage temperature data. From a different perspective, Fan *et al.* [28] use a particular method to forecast the price of electricity with *SOM* and *SVM*. Valero *et al.* [29] make classifications based on the first 8, 10 and 12 hours of day in order to compare the errors among them. Che *et al.* [30] use presents an adaptive fuzzy combination model based on the *SOM*, the *Support Vector Regression (SVR)* and the fuzzy inference method.

Regardless of the model used in the prediction, all works previously presented are applied to large areas or whole countries. Some other particular examples of forecasting in large areas with specific information about consumption values are the following ones: Chu *et al.* [31] handle values of 33000 MW to forecast the peak load; Wang *et al.* [32] manage high consumption in two large areas of China; on the other hand, Rejc and Partos [33] predict consumption between 1000 and 1500 MW in Slovenia, and Kebriaei *et al.* [34] show a large area of Iran with an approximate consumption of 1550 MW. Therefore, it was found necessary to extend this knowledge domain in order to study forecasting in smaller and less-aggregated environments which might have higher variability in the demand curve.

Demand forecasting is very important but not only in the context of large regions but also in disaggregated environments such as *microgrids*. That is why our work is focused on *microgrids*, a paradigm that is starting to be used in research but requires supporting tools - which still have to be developed- in order to reach its full potential. The work is completed with a study of error control in *ANN* models.

3. Computational Tools

This section shows the computational tools on which our architecture is based. In particular, the mathematical concepts of *SOM* model, clustering (*k-means* algorithm) and *MLP* model are presented.

3.1. *SOM*

SOM is one of the most popular *ANNs*; it was first described by Kohonen, who observed that in many regions of the brain cortex of higher animals there are areas where neurons that detect traits are distributed in a topological order (Kohonen [35]). The representation of the information in the brain cortex appears often organized spatially, and that is the real potential of *SOM* models (Kohonen [36,37]).

As shown in Figure 1, neurons are arranged in a two-layer architecture. The first is the input or sensory layer, consisting of m neurons (one for each input variable). The second layer forms a map of features and represents the space where processing occurs; this layer is usually square ($m \times m$), but it may be arranged in a different way.

Input neurons are labeled with the index k ($1 \leq k \leq m$), and $n_x \times n_y$ map neurons with a pair of indices $i \equiv (i, j)$ ($1 \leq i \leq n_x, 1 \leq j \leq n_y$), which determines its spatial location. Each input neuron (k) is connected to all the neurons (i, j) on the map by a synaptic weight (w_{ij}). In the learning phase, each neuron identifies particular features in the input space. Once an input vector $\mathbf{x}(t)$ is submitted and processed, the weight of winner neuron is modified in order to assimilate $\mathbf{x}(t)$. This process is repeated for a multitude of input vectors; when it is completed, the model identifies the specific domains (Voronoi domains) of the input space (Kohonen [38]).

3.2. Clustering: *k-means* algorithm

A Q clustering means partitioning a data set into a set of clusters $Q_i, i=1, \dots, C$; a widely adopted definition of optimal clustering is a partitioning that minimizes distances within and maximizes distances between clusters. An example of a commonly used algorithm is the *k-means*, which minimizes an error function (Vesanto and Alhoniemi [39]):

$$E = \sum_{k=1}^C \sum_{x \in Q_k} \|x - c_k\|^2 \quad (1)$$

where C is the number of clusters, and c_k is the center of k cluster. *k-means* is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters fixed *a priori*. The main idea is to define k centroids, one for each cluster.

In order to measure the strength of the associations of the potential clusters, a validation index is needed. In this work, we will use the Davies-Bouldin index (Davies and Bouldin [40]). This validation index is the most suitable for the evaluation of *k-means* partitioning since it provides low values, indicating good clustering results for spherical clusters, as some studies, such as Shim *et al.* [41], have demonstrated. Concretely, they compared several indexes.

k-means algorithm is currently being employed in different areas. For instance, Lee and Park [42] use *k-means* for image processing; Bishnu and Bhattacharjee [43] use *k-means* to forecast failures in software modules; on the other hand, Jiang *et al.* [44] use this algorithm for the detection of events in video images, and Nourbakhsh *et al.* [45] use it to identify customer types of unknown load profiles.

Vesanto and Alhoniemi [39] show that clustering of the *SOM* renders better results than clustering the data directly. The primary benefit of the two-level approach (*SOM*+clustering) is the reduction of the computational cost; even with relatively small number of samples, many clustering algorithms (specially hierarchical ones) become intractably heavy; for this reason, it is convenient to cluster a set of prototypes rather than clustering the data directly. Another benefit is noise reduction; after *SOM*, the prototypes are local averages of the data and, therefore, less sensitive to random variations than the original data.

3.3. MLP

Perceptron is a neural model that was introduced by Rosenblatt in the late fifties; its structure is based on the processing stages of sensory systems. *MLP* model -presented by Rumelhart (Rumelhart *et al.* [46])- has a topology of N layers, with off-line supervised learning and is

built by adding layers to a single *Perceptron*. *MLP* architecture (Figure 2) is used in economic environments to detect crisis, pattern recognition, image segmentation, data compression, applications in robotics, coding information, text translation of spoken language, applications in cardiology, etc.

The aim of the learning phase in *MLP* is to reach an optimal compromise in the adjustment of the weights in the different layers, using for this purpose the *backpropagation* algorithm. For minimizing the error the gradient descent method is used: a gradient to adjust the weights of the output layer (w'_{kj}) and a gradient to adjust the hidden layer (w_{ji}). In order to achieve this goal, pairs of input-target patterns are presented.

4. Architecture Model

4.1. Research data

The Spanish utility Iberdrola has facilitated the set of data necessary to do this research. This dataset includes historical information from 1 January 2008 to 31 December 2010, from the provincial capital of Soria (Castilla y León, Spain). The range of consumption varies from 7 to 39 MW, definitely much lower than values typically observed in a large aggregated environment. Its load curve presents all the features of that of a *microgrid*: moderate/small consumption figures, complex and highly variable shape, connected or disconnected to the grid.

From all the data, 70% were used for learning the global model, and the remaining 30%, for the prediction in the operation phase of the model.

4.2. Model

Hernández *et al.* [47] noted that all “*smart models*” (*Smart City*, *SG*, etc.) are aimed at achieving a set of common goals: sensing the environment, monitoring the operation and performing intelligent adaptation; the referenced work presents the umbrella term *Smart World*, meaning a framework for the interconnection and cooperation of several *Smart Systems*. The

authors present a study of the relationship between weather variables and electric load at *microgrid* scale, demonstrating that, even in a small scenario, this relationship still exists. *SGs*, systems capable of precise adaptation to particular local conditions, can greatly benefit from understanding the small-scale influence of weather (and other variables) in electric load. This is important because most of current *SG* deployments do not actually monitor weather variables nor employ them in their adaptation and prediction tasks. The correlation could potentially change with time in the same way that it has changed in the past; the place where the correlation tests are performed also influences the strength of this relationship, as it will be stronger in an environment with extreme weather than in a region with stable warm weather; therefore, as time and physical location are essential factors, applications should evolve with time, and focus on the site specific to the study, in order to obtain accurate information on the changing relationship between weather variables and electric power demand.

Most of the algorithms currently employed in power grids are designed to operate at a nationwide scale, and it is necessary to adapt them for operation in small environments (*microgrids*, *Smart Buildings*, etc.). This is especially true for load forecasting algorithms, and there is still a lot of work to do in order to be able to predict loads at small *microgrids* and even single nodes [48], but it is also true that new control algorithms have to be designed to operate at an extremely small scale. The study reported along [47] represents a very important step, since local weather variables and other parameters will be an extremely valuable input for node load/production forecast.

Hernández *et al.* [49] also confirm that demand forecasting models have focused on big regions or even entire countries. With the birth of the new energy environments (*microgrids* and *Smart Buildings*), the chance arises to conduct a more detailed study of the variables affecting electric load at smaller, localized areas.

The model developed consists of three stages as shown in Figure 3a. All the stages are based on the application of well-known algorithms, but their innovative combination in this complex architecture brings unique results. The first stage uses historical data (calendar and electricity consumption) to classify the patterns of different days using a *SOM*. In the second stage, the

SOM results are clustered using the *k-means* algorithm, in order to obtain the optimal neural classification. These two clustering stages operate in cascade in order to provide a better clustering than they can when operating on their own. The *SOM* is capable of identifying similarities among the different load patterns, and form different clusters. The *SOM* network uses the entire data set, it is not necessary to introduce the patterns with order. However, its output does not give information about how similar these clusters are one to another. Then, the *k-means* algorithm is capable of identifying these similarities among different clusters, and group similar clusters together. An easy explanation of the process is that the *SOM* operates grouping all the training patterns into $m \times n$ clusters. Each of these clusters is represented in the feature space by one “prototype” pattern which is effectively the “hyper-centroid” of the cluster, effectively reducing the dimension of the training sample that *k-means* has to face. Then, the *k-means* algorithm is used to cluster the *SOM* prototypes with better results. The input parameters given to *k-means* are the same ones than those used in the *SOM* network, completed with an additional variable which represents the number of the *SOM* neuron to which each pattern has been assigned in the first stage.

In the third stage, a dedicated *MLP* for each cluster is trained to properly forecast the load curve.

In the operation phase (once the networks have been trained), the cluster selector decides the appropriate cluster for the forecasting day, so its corresponding *MLP* is executed in order to obtain the forecast. Then, the necessary data is extracted from the database, including a set of variables that will be detailed later in this Section.

The complete algorithm is shown in Figure 3b. Its different sub-tasks are:

- *Data selection*: when the prediction starts a data selection process is activated, to connect with the database and retrieve the information from the different networks (see next step). Moreover, the process formats the dataset according to the learning strategy configured for each network.

- *Data in database*: the database contains raw data collected by several sensor networks (electric, weather, calendar, etc.). The database is therefore connected with different smart meters and dataloggers in order to obtain the data.
- *Data quality*: this process detects outliers with *Principal Component Analysis (PCA)*, as shown in [50]. If the retrieved data is labeled as outlier, it is discarded the process “*data selection*” is called again, so a new set of data is retrieved.
- *Creation of the SOM network*: the main parameters of the *SOM* used in this research are the following ones: the *SOM* dimension is 5×5 (25 neurons), which allows a good data dispersion; the shape of the neuron connections is hexagonal; the neighbourhood function employed is *Gaussian*; the initializing vector is *linear*; and the learning algorithm is a *batch* type.
- *Training of the SOM network*: the *SOM* network starts the learning stage, with the appropriate data. Electricity demand is dependent on the month, day of the week, workability and electricity demand from previous days, as shown in ([5], [13], [21], [29], [47] and [49-50]); therefore the number of inputs is 27, and the input variables of the *SOM* are, for each input patten: month (*January=1,...,December=12*); weekday (*Sunday=0,...,Saturday=6*); workability (*holiday=1* and *workingday=2*); and 24 values of hourly electricity consumption (daily load curve) of the day before the one to be forecasted.
- *k-means algorithm over SOM network*: the result of the *SOM*'s pattern recognition is used as an input to the *k-means* algorithm, whose configuration parameters are: maximum number of cluster, maximum number of trials and *k-means* algorithm batch type; the clustering is done in an unattended way. García-Lagos *et al.* [51] employ a *SOM* network for pattern recognition, then apply a manual clustering to create “*superclasses*” grouping several neurons together; the present work applies *k-means* after the *SOM* to perform this clustering task automatically.
- *Partitioning*: in a similar way to [5], the partition generated is studied in order to design a cluster selector, which allows the identification of the cluster one specific day belongs

to, according to its calendar parameters (Monday or Friday, Holiday or not, etc.); for example, a given cluster of days with low consumption is most likely to contain Sundays and holidays; then, the cluster selector is designed to assign Sundays and holidays to this cluster.

- *Creation of the MLP networks*: the *MLP* network is created. To optimize the topology of the *MLP*, we used a script varying all possible parameters, obtaining the following optimal configuration: the learning function used was the *Bayesian Backpropagation Regulation*; the function network performance was the *Sum Squared Error*; the number of entries in the network was 29; and the number of neurons of the hidden layer was 8, 6 and 10 in the clusters 1, 2 and 3 respectively.
- *Learning of the MLP network*: the *MLP* network starts the learning stage, with the appropriate data. In the *MLP* architecture (shown in Figure 4) periodic variables (day of the week and month) are supplied in the form of values of sines and cosines; it has been demonstrated that this transformation improves the performance, as shown by Drezga and Rahman [52] and Razavi and Tolson [53].

In a similar way to “*Learning of the SOM network*”, the model takes input values of electric consumption, day of the week and month of the day before the day to forecast; in addition, the estimation of the aggregated total consumption of the day to forecast is also fed into *MLP*. This input parameter list have been systematically optimized for this specific scenario and dataset. Therefore, variables in the input and output layer are the following ones:

- Input:
 - The electrical consumption is dependent on the consumption of the previous day, then, 24 values of the load curve ($L_{(d-1)1}, L_{(d-1)2}, \dots, L_{(d-1)24}$) of day $d-1$ are used (day before the forecasting day d).
 - The electrical consumption is dependent on the day of the week of the previous day, then, day of week (referring to $d-1$) is used, entered as two

different variables, in the form of *sines* and *cosines*, by means of $\sin[(2\pi day)/7](d-1)$ and $\cos[(2\pi day)/7](d-1)$.

- The electrical consumption is dependent on the month of the previous day, then, month (referring to $d-1$) is used, entered as two different variables, in the form of *sines* and *cosines*, by means of $\sin[(2\pi day)/12](d-1)$ and $\cos[(2\pi day)/12](d-1)$.
- Hernández *et al.* [54] present a two-stage model which first estimates some unknown values of the day to be forecasted (for example the aggregated demand -total load-) before estimating the hourly load. Following this model, Next Day's Total Load ($NDTL_d$) is used. Although there are many models available to estimate $NDTL_d$, as shown in Hsu and Chen [55], the aim of this paper is to present a simple architecture that uses a specific set of variables and yields excellent results; therefore, an error ranging $\pm 2\%$ with respect to the real value is added and the resulting values are used as input to validate the architecture proposed.
- Output: 24 values of the load curve (L_{d1}, \dots, L_{d24}) corresponding to the forecasting day (d).
- *Cluster selector*: cluster selector identifies the cluster to which the next day belongs (forecasting).
- *STLF with MLP*: with the data obtained (from the “*data in database*” process) and the selected cluster, the *MLP* network (*MLP* of the winner cluster) is executed to the forecast.

Complementing Figure 3b, the on-line scheme of operation of the predictor as shown in Figure 5 (to forecast the d day). The predictor selects the input data referring to $d-1$. In parallel, the cluster selector decides to which cluster belongs the d day. After this decision, the *MLP* of the selected cluster is executed along with the data selected by the predictor and the estimation of $NDTL_d$.

Formulas in [56] are employed to evaluate the performance of the model. During the operation phase, where the forecasting of d day is obtained, the *Mean Absolute Percentage Error* (*MAPE*) is calculated using:

$$MAPE_d = 100 \cdot \frac{\sum_{i=1}^{24} \left| \frac{L_d(i) - \hat{L}_d(i)}{L_d(i)} \right|}{24} \quad (2)$$

where $L_d(i)$ is the value measured for $t=i$ and d day, and $\hat{L}_d(i)$ is the estimated value.

The mean error of all the days of the operation phase is given by:

$$ERROR_{OP} = \frac{\sum_{i=1}^k MAPE_d}{k} \quad (3)$$

where k is the number of days to forecast.

The error for each of the 24 hours of the load curve in the operation phase is given by:

$$MAPE_{hi} = 100 \cdot \frac{\sum_{k=1}^n MAPE_{i,k}}{n} \quad (4)$$

where n is the size of the operation phase, $i=1, \dots, 24$, and $MAPE_{ik}$ is the error for i hour and k day.

5. Results

Along this section, the results of the operation phase are presented. The 70% of the data (without outliers) are used to be classified by *SOM*; the result of this classification is clustered by *k-means* algorithm, and finally the data belonging to each cluster are used to train a *MLP* linked to each cluster. The remaining 30% of the data will be used to obtain the prediction of load curves in the operation phase.

5.1. Pattern recognition with *SOM*

Figure 6 shows an activation scheme of the *SOM* map. In this Figure, hexagons represent neurons and links between neurons, so each neuron hexagon is surrounded by link hexagons representing connections with neighbor neurons. The result is that a 5×5 neuron grid is represented by a 9×9 hexagon grid, with five neurons in each row and column, and the rest of the cells representing the connections among them.

Then these hexagons are shaded to represent in addition the number of patterns classified by each neuron and the strength of each link. Finally, the image is colored to superimpose a specific example of the classification of holiday patterns (in red) vs. working days patterns (in blue): the bigger the colored neuron, the larger the number of patterns classified in that neuron. A visual analysis of the figure shows that holidays tend to be located in a particular area of the *SOM* (they are classified in neurons at the top-left corner of the map) and working days on a different one (classified in neurons at the bottom and right edges of the map).

After preliminary series of analyses, the following results were found: regarding workability, holidays are separated from the working days; referring to months, months patterns with similar seasonality are grouped in proximal neurons; respect to the day of the week, patterns linked to Saturdays and Sundays are grouped in close neurons, while the rest of the days do it in a different way.

5.2. Clustering with *k-means*

The results obtained by the different evaluations of the clusters with *k-means* show that the optimal number of clusters is three.

Once the clustering is done, it reminds visual conclusions that could be extracted after the phase of classify by *SOM*. The particularities of these clusters, including the number of neurons integrated in each of them are stated below:

- Cluster 1: working days of January, February, March, April, November, December and October (from 15th to 31st). Neurons: 3, 4, 5, 8, 9, 10, 13, 14, 15 and 20.
- Cluster 2: Saturdays, Sundays and Holidays. Neurons: 1, 2, 6, 7, 11, 12 and 16.
- Cluster 3: working days of May, June, July, August, September and October (from 1st to 14th). Neurons: 17, 18, 19, 21, 22, 23, 24 and 25.

Figure 7 shows all the curves belonging to each of the three clusters; it is easy to identify the high similarity between curves of the same cluster, and the appreciable difference when compared with those of the other clusters.

5.3. General results

Once the three *MLP* networks have been trained with the set of patterns corresponding to each cluster, the operation phase is performed. The days to forecast (*d*) pass through the cluster selector, and it decides to which cluster belongs *d*.

For each day to forecast, the load curve and its mean error of the day is calculated with (2). For example, in Figure 8 the evolution of errors in cluster 1 is represented with the *mean*, *mean ± standard deviation* and *mean ± 2x standard deviation*. On the vertical axis, the error per unit is shown and the abscissa axis represents the forecasting day. The labels on the graphic, for example “2 4/1 - 4 1”, can be interpreted as follows: “2” represents workability of the previous day; “4 /” corresponds to the month of the day to forecast (April); “/ 1” represents the day of the month to forecast; “4” designates the day of the week (Thursday) of the forecasting day; and “1” is workability of the forecasting day. The mean error in the operation phase for

each cluster is obtained using (3), being 2.7615% for cluster 1, 3.1755% for cluster 2 and 2.7159% for cluster 3.

Most of the errors occur in the first interval, as shown in Table 1, in which the number of days with error is represented in the following ranges: *mean* and *mean±standard deviation*; *mean* and *mean±2xstandard deviation*; over the *standard deviation* and over the *2xstandard deviation*.

Figure 9 shows the errors per hours calculated with (4); for cluster 3, the highest percentage of error is shown in specific segments of the load curve (the two points with the lowest consumption and the two points with the highest consumption). The data corresponding to the error per hour (4) of each cluster are 2.7615% in cluster 1, 3.1755% in cluster 2 and 2.7159% in cluster 3.

Figure 10 represents the forecasting curves of three days with a high percentage of error, each one belonging to a different cluster, where: a) 12/24/2010 (cluster 1) with an error of 4.529%; b) 6/24/2010 (cluster 2) with an error of 4.42%; and c) 6/28/2010 (cluster 3) with an error of 5.1361%. The red curve in the figure shows the real curve (the curve that wants to be forecasted); the black one outlines the forecasting done and the green curves represent all the curves similar to the day that is forecasted (same day of the week and workability).

On the other hand, Figure 11 shows the forecasting curves of three days with a low percentage of error (each belonging to a different cluster), where: a) 12/21/2010 (cluster 1) shows an error of 1.0925%; b) 7/24/2010 (cluster 2) presents an error of 1.6733%; and c) 8/25/2010 (cluster 3) shows an error of 1.4844%.

6. Analysis of Results

6.1. Distribution of errors

As seen in Figure 8 and Table 1, mean errors of the day are included in the *mean±standard deviation* range, corresponding to errors between 4.01%–1.51% for the cluster 1, 4.29%–2.05% for cluster 2 and 3.89%–1.53% for cluster 3. These ranges can be considered acceptable.

As seen in Figure 9 and Figure 10, the mean error per hours shows that the highest errors correspond to particular zones of the load curve for all clusters. Therefore, it is possible to use some information of the load curve to forecast, in connection with the aforementioned zones, which enables a more accurate prediction.

6.2. Errors per day of week and month

Figures 12, 13, and 14 show the evolution of mean error per days in every cluster (it appears per day of the week). It is necessary to emphasize that in cluster 2 (Saturdays, Sundays and holiday), holidays (not weekend) represent a very small number of days; that is why it has not been included in the figure. In all the clusters, every Friday (beginning of the weekend) normally has a behavior with less consumption than the rest of working days (from Monday to Thursday); for this reason, its mean error can be slightly higher in mean than that of the rest of the days. In clusters 1 and 3, the highest mean errors correspond to Friday, for the same reason mentioned above, and Monday tends to have lower consumption patterns because of the inertia of the weekend; that why, they are days more complicated to forecast with the information of the cluster. In cluster 2, the high errors of holiday days from Monday to Friday can be explained by the low number of patterns with similar characteristics in the learning phase.

If the same study is done for months, mean error per month changes from 3.5302% to 2.199% in cluster 1, from 3.8811% to 2.3646% in cluster 2 and from 3.3739% to 2.2385% in cluster 3. From the monthly point of view, the forecasting can be considered optimum.

6.3. Dependence of the error with the number of learning patterns

It is important to understand the evolution of the error (operation phase – *MAPE*) with respect to the number of learning patterns. The aim of this section is therefore to know if there is a chance to get a sensible improvement on the learning phase using a larger dataset for training, and therefore to understand which is a good size of the training dataset. With this size, a “sliding window” can be implemented in order to always employ the fresher data available (to account for the dynamicity of the system) while still retaining enough historical data to properly train the forecaster.

Hamamoto *et al.* [57] show the effect of the learning patterns with respect to the number of hidden layer neurons, paying special attention to the results when the number of features is large or when the number of learning patterns is small; that's why Raudys and Jain [58] emphasize the importance of paying attention to design. Blamire [59] explores the impact of the relative size of the sample sets used to define candidate classes in order to improve accuracy in the classification process using *ANN* techniques, suggesting that the network uses a lot of time in the formation of classes. Foody *et al.* [60] combine these results to use them in the imaging area, or in the area of medicine, as Figueroa *et al.* [61] show.

For these reasons, it was considered very important to demonstrate the dependence of the mean error (operating phase) with the number of learning patterns (for each cluster), taking into account that network topology varies with the number of patterns. Table 2 shows the improvement of error percentage at the beginning of cluster 3 versus the other two clusters, because load curves in cluster 3 are more similar to each other.

Figure 15 shows the evolution of the error versus the number of patterns for each cluster. Later, results are adjusted with the best fitting function for each cluster, the functions are:

$$Y = 42.0149 \cdot X^{(-0.48474)} \quad (5)$$

$$Y = 18.6268 \cdot X^{(-0.32758)} \quad (6)$$

$$Y = 3.9005 + 3.9005 \cdot X - 0.012207 \cdot X^2 \quad (7)$$

where Y is the mean error in the operation phase and X represents the number of patterns in the learning phase; (5) corresponds to the cluster 1 (Figure 15a), (6) to the cluster 2 (Figure 15b) and (7) to the cluster 3 (Figure 15c). To compare the improvement, an architecture has been developed with a single *MLP* and the same variables. This model is trained with 730 patterns and an operation phase similar to the one presented in this work. Figure 15d represents the evolution of the error with respect to the number of patterns. *MLP* model has an error of 3.50% compared to 2.78% of cluster 1 (for 275 patterns); an error of 4.06% compared to 3.19% of cluster 2 (for 225 patterns); and an error of 4.27% compared to 2.73% of cluster 3 (for 200 patterns).

Figure 15d shows that *MLP* does not have a sensible tendency to decrease the error percentage above 700 patterns; however, the three clusters (sequence *SOM+k-means+MLP*) do have a tendency to improve, given that the test has a number of 275, 225 and 200 learning patterns for each one of the three clusters. Therefore, it has been demonstrated the improvement of the classification and subsequent clustering to forecast load curves, since each cluster is specialized in similar curves, obtaining smaller forecasting error. With 200 patterns, if 25 patterns are added, the improvement is: 0.50% (cluster 1), 0.14% (cluster 2) and 0.10% (cluster 3); the improvement of the model *MLP* (Figure 15d) would be 0.02%.

Chan *et al.* [62] present a *STLF* model for a *microgrid*, using a *Multiple Classifier Systems (MCS)*, with a set of learning data similar to the set used in this work, and a load curve with a peak of 2 MW, but with a phase operation of 20 days (shorter than the one presented here). The *MAPE* of this model for the operation phase is 15.12%, doing a combination with *Radial Basis Function Neural Network (RBFNN)*; and 15.66%, doing a combination with *Generalized Regression Neural Network (GRNN)*. In both models, *MAPE* is higher than even in the worst of our clusters (3.1755%), so we can conclude that our model is more efficient; moreover, it is necessary also to keep in mind that the number of patterns used along this work is much lower than the number used in other models.

An *et al.* [63] propose a novel approach that combines a *MLP* with empirical mode decomposition based signal filtering and seasonal adjustment. The *MAPE* of this model for the operation phase is 1.07%-4.79%, *MAPE* is higher than even in the worst of our clusters. The work is applied to a large area, New South Wales.

7. Conclusions and Future Works

Microgrids are a reality, being physical spaces characterized by small electricity consumption compared to larger provinces or countries. Therefore, the development of new models based on *ANN* for *STLF* appears to be very attractive and show a margin of improvement.

In this paper, a model based on three levels has been presented: a pattern recognition to classify with *SOM*, a clustering with *k-means* algorithm, and finally a *MLP* model for each of the

clusters obtained in the clustering process. The model produces low errors compared to other simple models that are not specialized by means of classification and clustering. Moreover, a methodology to control the error evolution has been presented and shows the dependence of error (operation phase) on the number of patterns (learning phase), which has led to the usage of a “sliding window” for system training.

In future works we will include new demand prediction models and generation in disaggregated environments. Several factors such as predictions using the information of the curve to be predicted will be added. Moreover, we will try to validate our system in other *microgrid*-sized environments. Continuing with the work of Marinescu *et al.* [64], we will prove local variables (for example the climatic ones of the *microgrid*). Marinescu *et al.* [64] demonstrate that a simple on-demand water heater (5 kW) switched on at a different time of the day can shift the overall demand with more than 10% for one hour, therefore, climatic variables must be employed in the models, therefore, climatic variables must be employed in the models. Tomic [65] focuses on the impact of forecasting errors on the economic effects of trading and balancing in the *microgrid*: while lower forecasting accuracy induces greater differences between forecasted and real consumption/generation, and hence higher need for balancing energy, higher forecasting precision may increase the cost of the system; the *Microgrid System Operator (MSO)* operates the local market and balances demand and supply.

Acknowledgments

We would like to thank Iberdrola, specially to Óscar Villanueva, Head of Business Exploitation in Burgos and in Soria, and Silvia Herrero, Head of Communication of Iberdrola in Castilla y León for the information provided, and Javier Sanjuán and Alvaro González from the University of Zaragoza for their collaboration with this study.

References

- [1] European Commission, 2010. Towards a low-carbon future. http://ec.europa.eu/energy/technology/set_plan/set_plan_en.htm. Brochure. Accessed 30 June 2012.
- [2] Tang F, Li M, Huang JC. Real-time transaction processing for autonomic Grid applications. *Engineering Applications of Artificial Intelligence* 2004;17(7):799-807.
- [3] Lasseter R, Akhil A, Marnay C, Stevens J, Dagle J, Guttromson R, Meliopoulos AS, Yinger R, Eto J. White paper on integration of Distributed Energy resources. The CERTS Microgrid concept. Consortium for Electric Reliability Technology Solutions, prepared for the US department of Energy 2002.
- [4] Venayagamoorthy GK, Welch RL. Energy dispatch controllers for a photovoltaic system. *Engineering Applications of Artificial Intelligence* 2010;23(2):249-61.
- [5] Hernández L, Baladrón C, Aguiar JM, Carro B, Sánchez-Esguevillas A. Classification and Clustering of Electricity Demand Patterns in Industrial Parks. *Energies* 2012;5(12):5215-28.
- [6] Abu-Shikhah N, Elkarmi F. Medium-term electric load forecasting using singular value decomposition. *Energy* 2011;36(7):4259-71.
- [7] Ekonomou L. Greek long-term energy consumption prediction using artificial neural networks. *Energy* 2010;35(2):512-17.
- [8] Park DC, El-Sharkawi MA, Marks RJ II, Atlas LE, Damborg MJ. Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems* 1991;6(2):442-49.
- [9] Ho K-L, Hsu Y-Y, Yang C-C. Short term load forecasting using a multilayer neural network with and adaptative learning algorithm. *IEEE Transactions on Power Systems* 1992;7(1):141-49.
- [10] Ho K-L, Hsu Y-Y, Chen C-F, Lee T-E, Liang C-C, Lai T-S, Chen K-K. Short term load forecasting of Taiwan power system using a knowledge-based expert system. *IEEE Transactions on Power Systems* 1990;5(4):1214-21.
- [11] Dash PK, Satpathy HP, Liew AC. A real-time short-term peak and average load forecasting system using a self-organising fuzzy neural network. *Engineering Applications of Artificial Intelligence* 1988;11(2):307-16.
- [12] Drezga I, Rahman S. Input variable selection for ANN-based short-term load forecasting. *IEEE Transactions on Power Systems* 1998;13(4):1238-44.

- [13] Drezga I, Rahman S. Short-term load forecasting with local ANN predictors. *IEEE Transactions on Power Systems* 1999;14(3):844-50.
- [14] McMenemy JS, Monforte FA. Short-term energy forecasting with neural networks. *The Energy Journal* 1998;19(4):43-61.
- [15] Lee KY, Cha YT, Park JH. Short-term load forecasting using an artificial neural network. *IEEE Transactions on Power Systems* 1992;7(1):124-32.
- [16] Lu C-N, Wu H-T, Vemuri S. Neural network based short term load forecasting. *IEEE Transactions on Power Systems* 1993;8(1):336-42.
- [17] Papalexopoulos AD, Hao S, Peng TM. An implementation of a neural network based load forecasting models for the EMS. *IEEE Transactions on Power Systems* 1994;9(4):1956-62.
- [18] Bakirtzis AG, Petridis V, Kiartzis SJ, Alexiadis MC, Maissis AH. A neural network short term load forecasting model for the Greek power system. *IEEE Transactions on Power Systems* 1996;11(2):858-63.
- [19] Lou CW, Dong MC. Modeling data uncertainty on electric load forecasting based on Type-2 fuzzy logic set theory. *Engineering Applications of Artificial Intelligence* 2012;25(8):1567-76.
- [20] Deihimi A, Showkati H. Application of echo state networks in short-term electric load forecasting. *Energy* 2012;39(1):327-40.
- [21] Marín FJ, García-Lagos F, Joya G, Sandoval F. Global model for short-term load forecasting using artificial neural networks. *IEE Proceedings-Generation, Transmission and Distribution* 2002;149(2):121-5.
- [22] Joya G, García-Lagos F, Atencia M, Sandoval F. Artificial Neural Networks for Energy Management System. Applicability and Limitations of the Main Paradigms. *European Journal of Economic and Social System* 2004;17(1-2):11-28.
- [23] Mori H, Itagaki T. A Precondition Technique with Reconstruction of Data Similarity Based Classification for Short-term Load Forecasting. *IEEE Power Engineering Society General Meeting*, 6-10 Jun. 2004, Denver (Colorado, USA);1:280-5.
- [24] Carpinteiro OAS, Reis AJR. A SOM-based hierarchical model to short-term load forecasting. *IEEE Russia Power Tech*, 27-30 Jun. 2005, St. Petersburg (Russia);1-6.
- [25] Wang ZY. Development Case-based Reasoning System for Short-term Load Forecasting. *IEEE Russia Power Engineering Society General Meeting*, 16 Oct. 2006, Montreal (Quebec, Canada);1-6.

- [26] Fan S, Chen L. Short-Term Load Forecasting Based on an Adaptive Hybrid Method. *IEEE Transactions on Power Systems* 2006;21(1):392-401.
- [27] Farhadi M, Tafreshi SMM. Effective Model for Next Day Load Curve Forecasting Based Upon Combination of Perceptron and Kohonen ANNs Applied to Iran Power Network. 29th International Telecommunications Energy Conference, 30 Sep.-4 Oct. 2007, Rome (Italy);267-73.
- [28] Fan S, Mao C, Chen L. Next-day electricity-price forecasting using a hybrid network. *IET Generation Transmission & Distribution* 2007;1(1):176-82.
- [29] Valero S, Aparicio J, Senabre C, Ortiz M, Sancho J, Gabaldon A. Analysis of Different Testing Parameters in Self-Organizing Maps for Short-Term Load Demand Forecasting in Spain. *Proceedings of the International Symposium on Modern Electric Power Systems*, 20-22 Sep. 2010, Wroclaw (Poland);1-6.
- [30] Che J, Wang J, Wang G. An adaptive fuzzy combination model based on self-organizing map and support vector regression for electric load forecasting. *Energy* 2012;37(1):657-64.
- [31] Chu W-C, Chen Y-P, Xu Z-W, Lee W-J. Multiregion Short-Term Load Forecasting in Consideration of HI and Load/Weather Diversity. *IEEE Transactions on Industry Applications* 2011;47(1):232-7.
- [32] Wang Y, Xia Q, Kang C. Secondary Forecasting Based on Deviation Analysis for Short-Term Load Forecasting. *IEEE Transactions on Power Systems* 2011;26(2):500-7.
- [33] Rejc M, Partos M. Short-Term Transmission-Loss Forecast for the Slovenian Transmission Power System Based on a Fuzzy-Logic Decision Approach. *IEEE Transactions on Power Systems* 2011;26(3):1511-21.
- [34] Kebriaei H, Araabi BN, Rahimi-Kian A. Short-Term Load Forecasting With a New Nonsymmetric Penalty Function. *IEEE Transactions on Power Systems* 2011;26(4):1817-25.
- [35] Kohonen T. The Self-organizing Map. *Proceedings of the IEEE* 1990;78(9):1464-80.
- [36] Kohonen T. Analysis of a simple self-organizing process. *Biological Cybernetics* 1982;44(2):135-40.
- [37] Kohonen T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 1982;43(1):59-69.
- [38] Kohonen T. The neural phonetic typewriter. *IEEE Computer Magazine* 1988;21(3):11-22.

- [39] Vesanto J, Alhoniemi E. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks* 2000;11(3):586-600.
- [40] Davies DL, Bouldin DW. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 1979;1(2):224-7.
- [41] Shim Y, Chung J, Choi I-C. A Comparison Study of Cluster Validity Indices Using a Nonhierarchical Clustering Algorithm. *International Conference on Computational Intelligence for Modeling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Nov. 2005, Vienna (Austria)*;199-204.
- [42] Lee JW, Park R-H, Chang S. Local tone mapping using the K-means algorithm and automatic gamma setting. *IEEE Transactions on Consumer Electronics* 2011;57(1):209-17.
- [43] Bishnu PS, Bhattacharjee V. Software Fault Prediction Using Quad Tree-Based K-Means Clustering Algorithm. *IEEE Transactions on Power Systems* 2012;24(6):1146-50.
- [44] Jiang Z, Lin Z, Davis LS. Recognizing Human Actions by Learning and Matching Shape-Motion Prototype Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2012;34(3):533-47.
- [45] Nourbakhsh G, Eden G, McVeigh D, Ghosh A. Chronological Categorization and Decomposition of Customer Loads. *IEEE Transactions on Power Delivery* 2012;27(4):2270-77.
- [46] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by backpropagation errors. *Nature* 1996;323:533-6.
- [47] Hernández L, Baladrón C, Aguiar JM, Calavia L, Carro B, Sánchez-Esguevillas A, Cook DJ, Chinarro, D, Gómez J. A Study of the Relationship between Weather Variables and Electric Power Demand inside a Smart Grid/Smart World Framework. *Sensors* 2012, 12(9):11571-11591.
- [48] Hernández L, Baladrón C, Aguiar JM, Carro B, Sánchez-Esguevillas A, Lloret J, Massana J. A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings. *IEEE Communications Surveys and Tutorials* 2014, pending publication (January).
- [49] Hernández L, Baladrón C, Aguiar JM, Calavia L, Carro B, Sánchez-Esguevillas A, García P, Lloret J. Experimental Analysis of the Input Variables' Relevance to Forecast Next Day's Aggregated Electric Demand Using Neural Networks. *Energies* 2013, 6(6):2927-2948.

- [50]Hernández L, Baladrón C, Aguiar JM, Carro B, Sánchez-Esguevillas A, Lloret J. Short-Term Load Forecasting for Microgrids Based on Artificial Neural Networks. *Energies* 2013, 6(3):1385-1408.
- [51]García-Lagos, F. Joya, G, Marín FJ, Sandoval F. Global model for short-term load forecasting using artificial neural networks. *IET Proceedings - Generation Transmission and Distribution* 2002, 149(2):121-125.
- [52]Drezga I, Rahman S. Phase-space based short-term load forecasting for deregulated electric power industry. *Proceedings of International Joint Conference on Neural Networks*, 10-16 July 1999, Washington DC (US);5:3405-9.
- [53]Razavi S, Tolson BA. A New Formulation for Feedforward Neural Networks. *IEEE Transactions on Neural Networks* 2011;22(10):1588-98.
- [54]Hernández L, Baladrón C, Aguiar JM, Calavia L, Carro B, Sánchez-Esguevillas A, Sanjuán J, González A, Lloret J. Improved Short-Term Load Forecasting Based on Two-Stage Predictions with Artificial Neural Networks in a Microgrid Environment. *Energies* 2013, 6(9):4489-4507.
- [55]Hsu C-C, Chen C-Y. Regional load forecasting in Taiwan—applications of artificial neural networks. *Energy Conversion and Management* 2003;44(12):1941-9.
- [56]Hernández L, Baladrón C, Aguiar JM, Calavia L, Carro B, Sánchez-Esguevillas A, Sanjuán J, González A, Lloret J. Improved Short-Term Load Forecasting Based on Two-Stage Predictions with Artificial Neural Networks in a Microgrid Environment. *Energies* 2013;6(9):4489-4507.
- [57]Hamamoto Y, Uchimura S, Kanaoka T, Tomita S. Evaluation of artificial neural network classifiers in small sample size situations. *Proceedings of 1993 International Joint Conference on Neural Networks*, 25-29 Oct. 1993, Nagoya (Japan);1733-5.
- [58]Raudys SJ, Jain AK. Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1991;13(3):252-64.
- [59]Blamire PA. The influence of relative sample size in training artificial neural networks. *International Journal of Remote Sensing* 1996;17(1):223-30.
- [60]Foody GM, McCulloch MB, Yates WB. The effect of training set size and composition on artificial neural network classification. *International Journal of Remote Sensing* 1995;16(9):1707-23.

- [61] Figueroa RL, Zeng-Treitler Q, Kandula S, Ngo LH. Predicting sample size required for classification performance. *BMC Medical Informatics & Decision Making* 2012;12(8):1-10.
- [62] Chan PPK, Chen W-C, Ng WWY, Yeung DS. Multiple Classifier System for Short Term Load Forecast of Microgrid. *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics*, 10-13 July 2011, Guilin (China);1268-73.
- [63] An N, Zhao W, Wang J, Shang D, Zhao E. Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting. *Energy* 2013;49(1):279-88.
- [64] Marinescu A, Harris C, Dusparic I, Clarke S, Cahill V. Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods. *2013 2nd International Workshop on Software Engineering Challenges for the Smart Grid (SE4SG)*, 10-18 May 2013, San Francisco CA (US); 25-32.
- [65] Tomic SD. A Study of the Impact of Load Forecasting Errors on Trading and Balancing in a Microgrid. *2013 IEEE Green Technologies Conference*, 4-5 April 2013, Denver CO (US); 443-50.

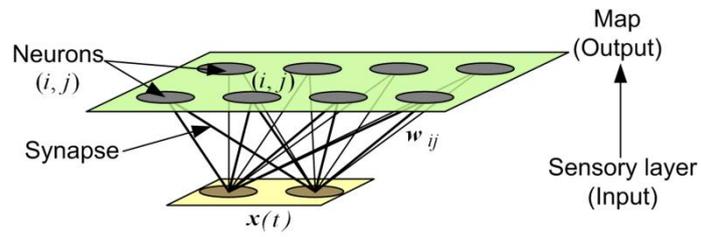


Figure 1. SOM architecture.

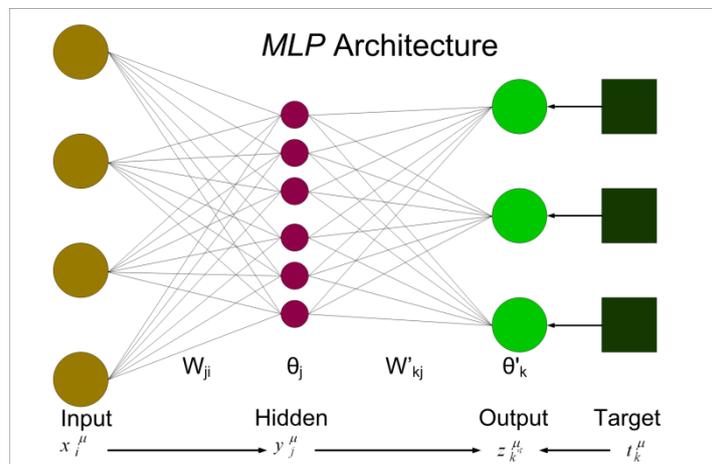


Figure 2. MLP architecture scheme: x_i represents the input of the network, y_i is the output of the hidden layer, z_k is the output of the output layer, t_k represents the target output, w_{ji} is the weight of the hidden layer, θ_j is the threshold of the hidden layer, w'_{kj} is the weight of the output layer and θ'_k is the threshold of the output layer.

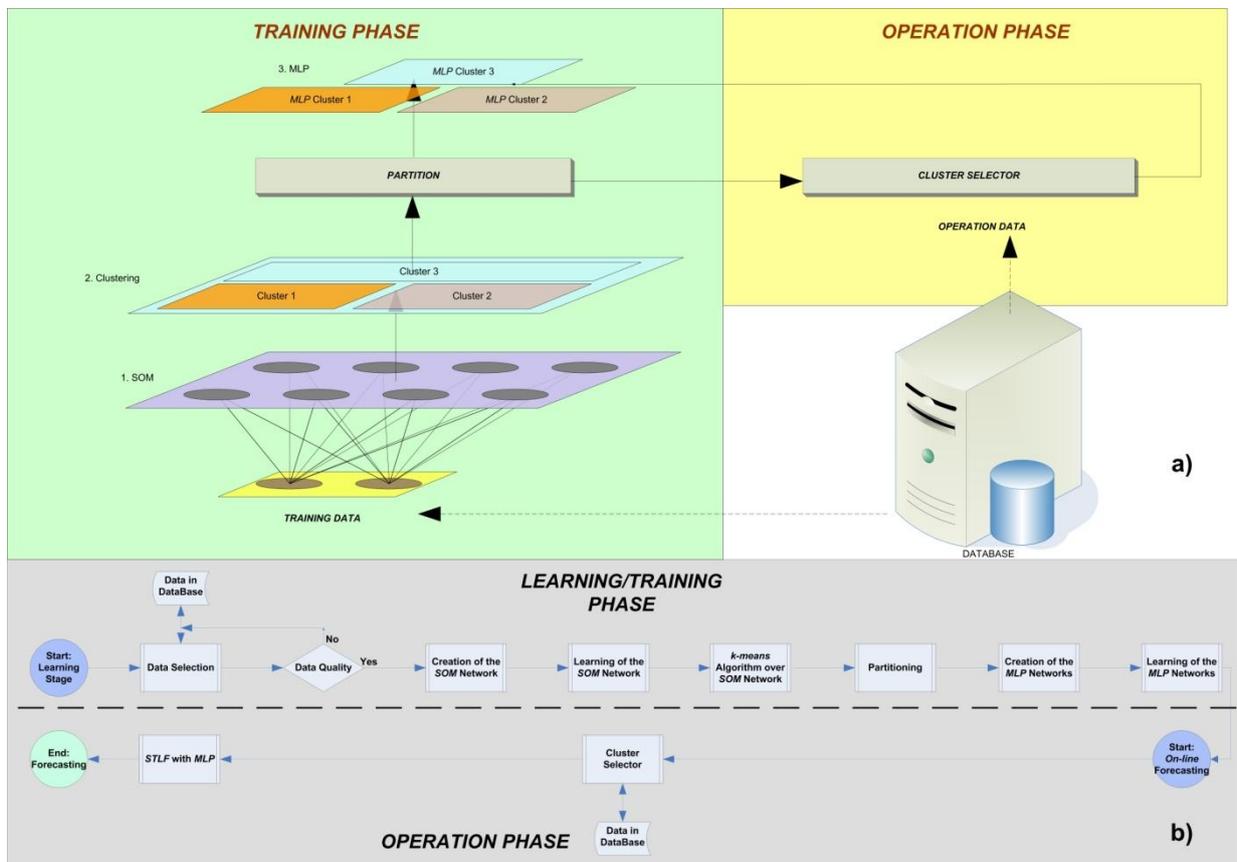


Figure 3. a) General model; b) Complete algorithm

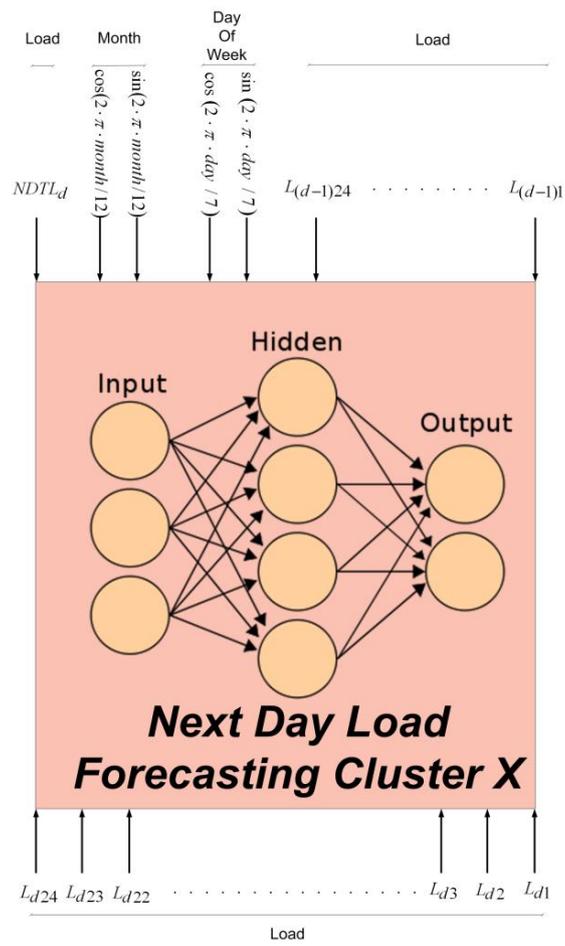


Figure 4. MLP architecture.

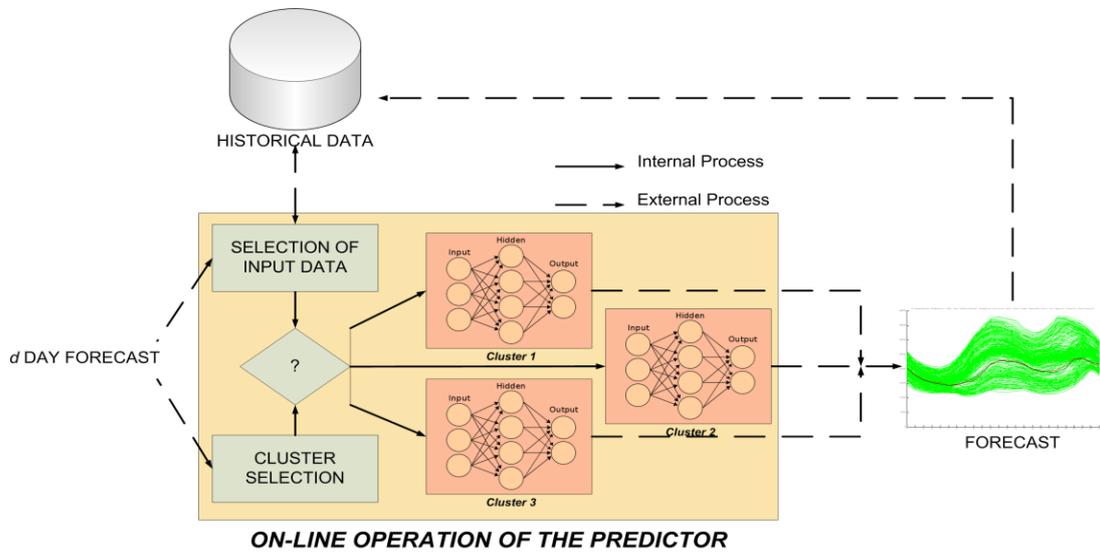


Figure 5. On-Line operation of the predictor.

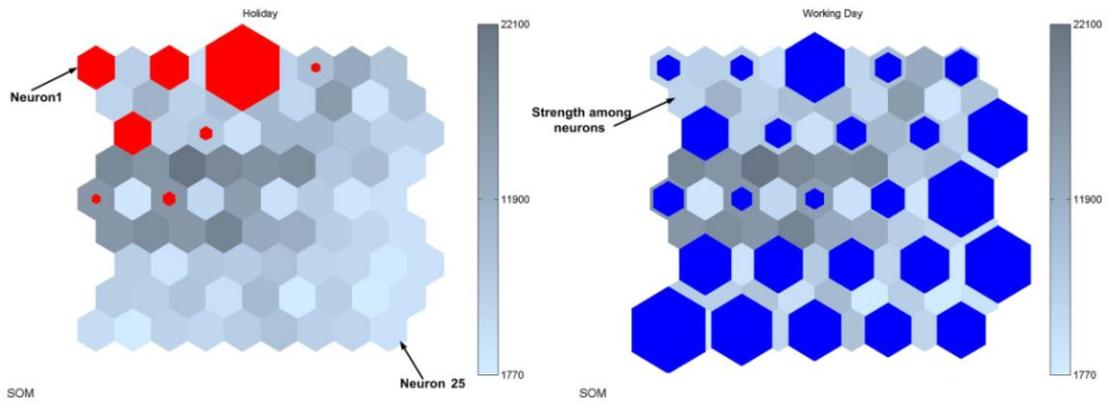


Figure 6. Activation map for workability.

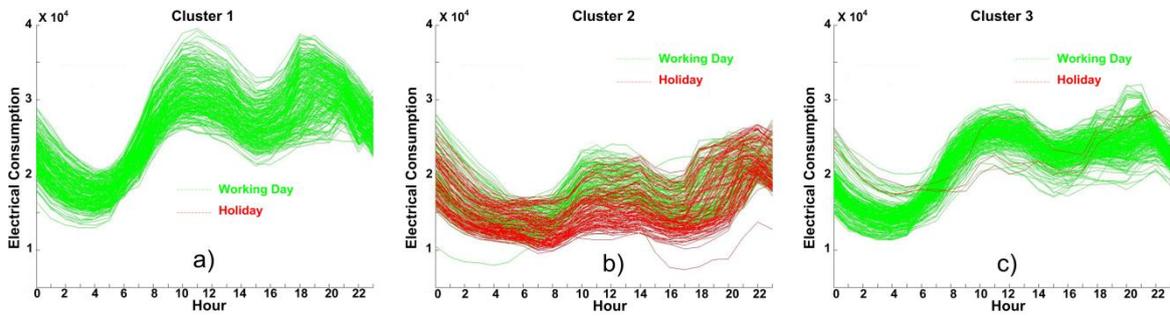


Figure 7. Load curves of three clusters: a) cluster 1; b) cluster 2; c) cluster 3.

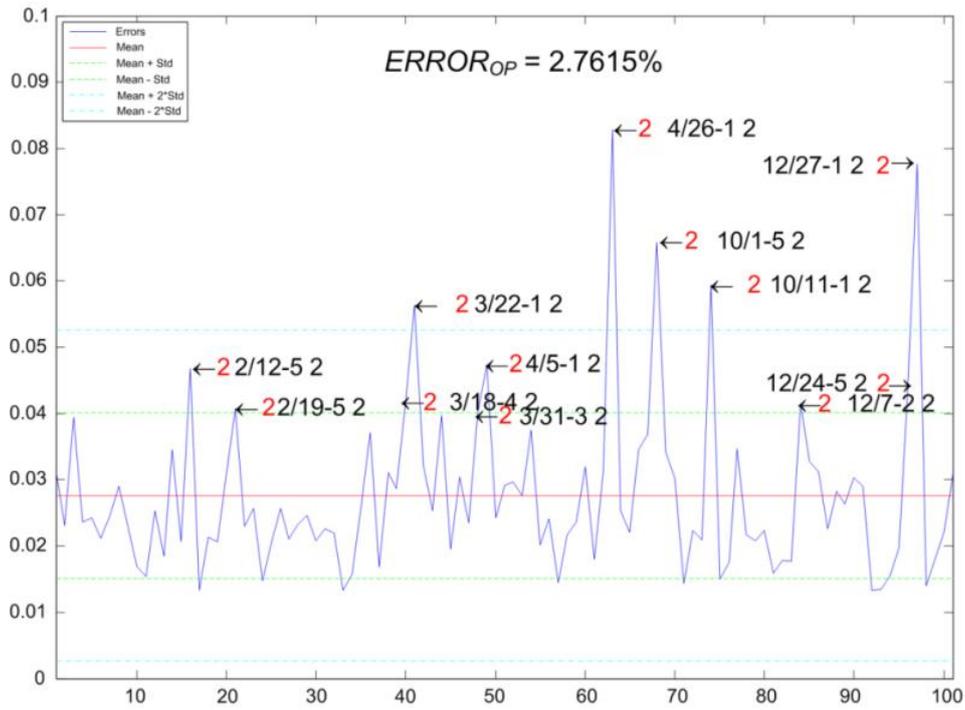


Figure 8. Errors per day in cluster 1.

Table 1. Distribution of errors per days.

CLUSTER 1		CLUSTER 2		CLUSTER 3	
Variable	%	Variable	%	Variable	%
Mean	2.76	Mean	3.17	Mean	2.71
Std. Des.	1.25	Std. Des.	1.12	Std. Des.	1.18
Over x1 Std.	11.88	Over x1 Std.	13.97	Over x1 Std.	13.97
Between x1 Std.	81.18	Between x1 Std.	78.49	Between x1 Std.	76.34
Under x1 Std.	6.69	Under x1 Std.	7.52	Under x1 Std.	9.67
Over x2 Std.	4.95	Over x2 Std.	3.22	Over x2 Std.	5.37
Between x2 Std.	95.04	Between x2 Std.	96.77	Between x2 Std.	94.62
Under x2 Std.	0.00	Under x2 Std.	0.00	Under x2 Std.	0.00

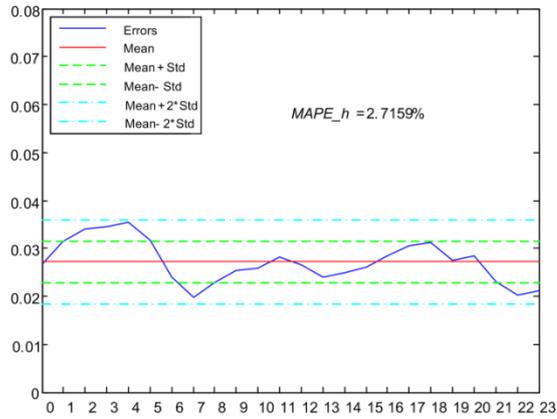


Figure 9. Errors per hours in cluster 3.

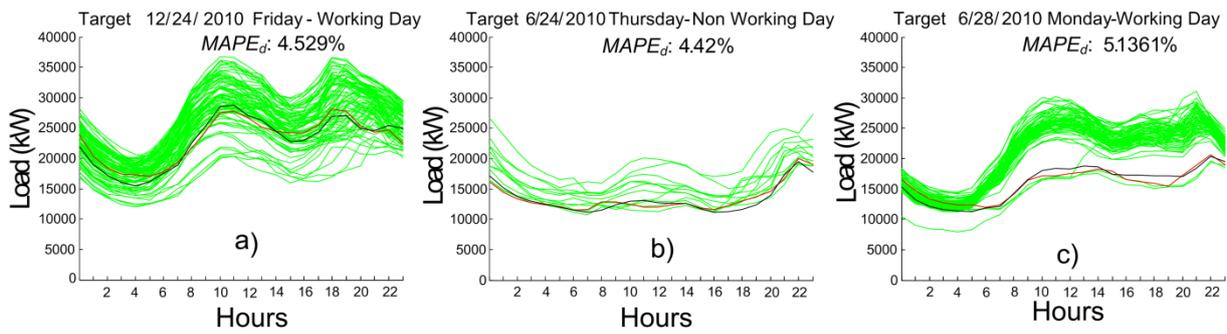


Figure 10. Forecasting with high errors: a) cluster 1: 12/24/2010; b) cluster 2: 6/24/2010; c) cluster 3: 6/28/2010.

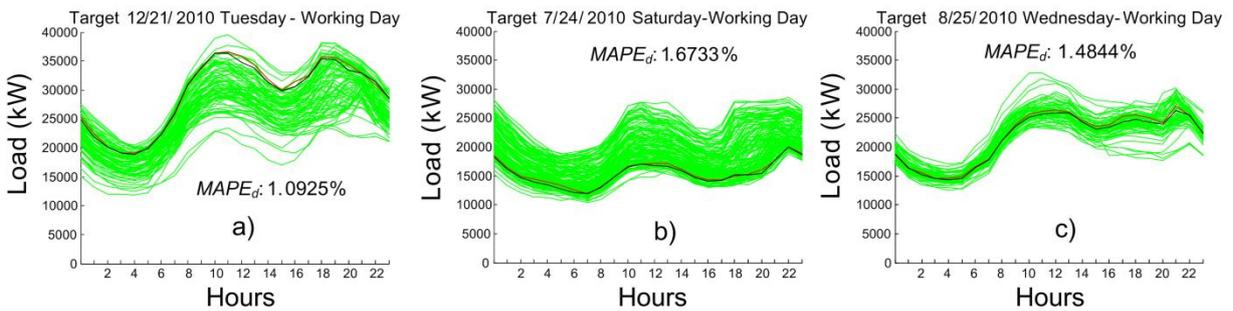


Figure 11. Forecast with low errors: a) cluster 1: 12/21/2010; b) cluster 2: 7/24/2010; c) cluster 3: 8/25/2010.

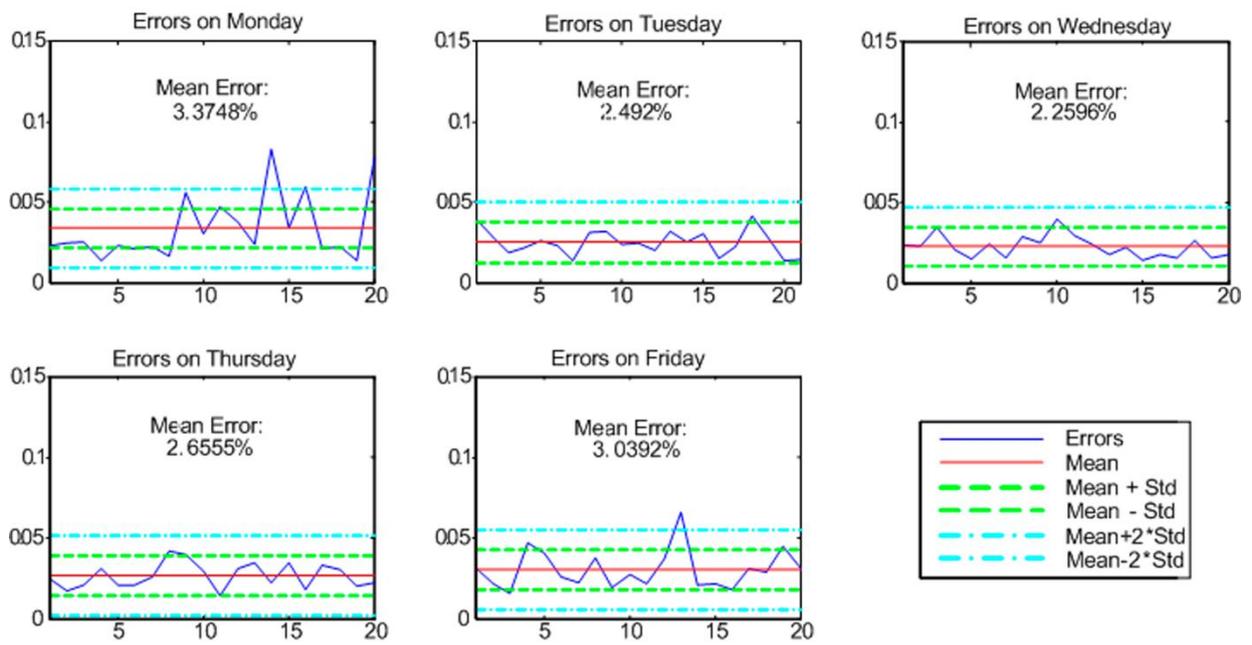


Figure 12. Errors per days: cluster 1. The vertical axis shows the forecasting error (per unit), and the abscissa axis represents the number of days forecasted for that day of the week.

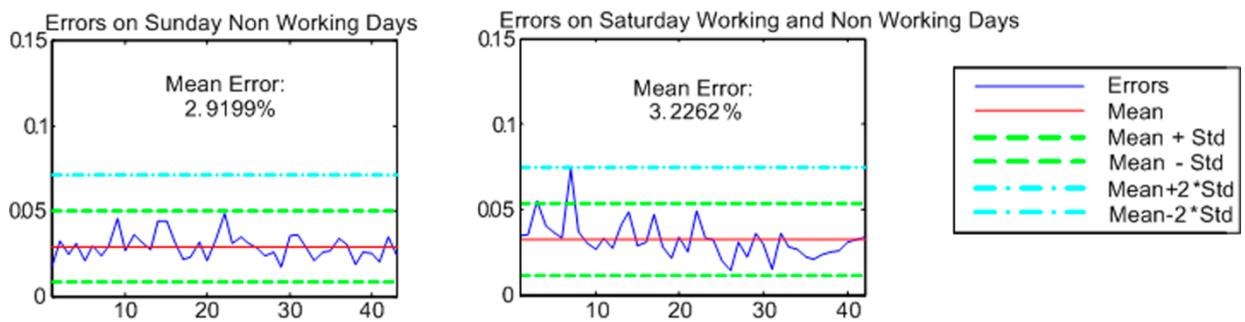


Figure 13. Errors per days: cluster 2. The vertical axis shows the forecasting error (per unit), and on abscissa axis the number of days forecasted for that day of the week is represented.

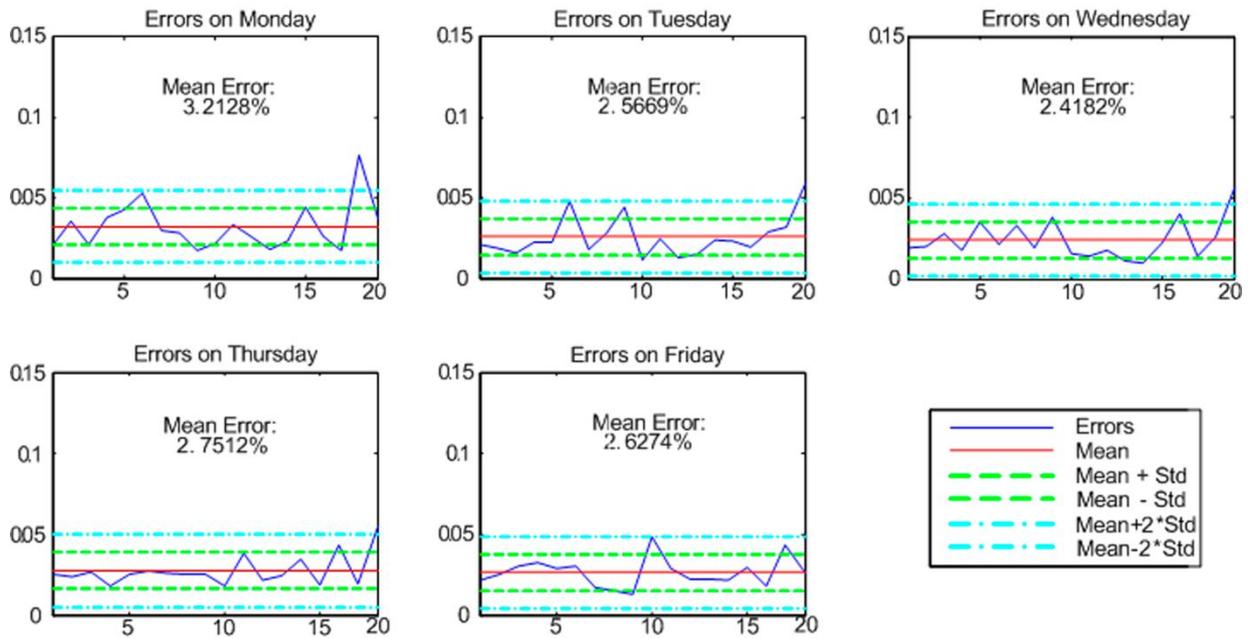


Figure 14. Errors per days: cluster 3. The vertical axis shows the forecasting error (per unit), and on abscissa axis each the number of days forecasted for that day of the week is represented.

Table 2. Dependence of the error on the number of learning patterns. *NN*: network number; *Ptt*: patterns; *N*: number of neurons; *MAPE*(%): mean error of total days in the operation phase; *Std. Desv.* (%): standard deviation of the operation phase (*MAPE*).

CLUSTER 1					CLUSTER 2					CLUSTER 3				
NN	Ptt	N	MAPE	Std. Desv	NN	Ptt	N	MAPE	Std. Desv	NN	Ptt	N	MAPE	Std. Desv
1	50	3	9.63	7.13	1	50	3	7.58	5.60	1	50	3	4.50	2.55
2	75	3	8.13	6.15	2	75	3	6.58	4.52	2	75	4	3.59	1.63
3	100	4	5.53	3.45	3	100	4	5.39	3.29	3	100	4	2.90	1.45
4	125	5	4.11	2.25	4	125	4	3.86	1.84	4	125	6	2.87	1.32
5	150	6	4.10	2.21	5	150	4	3.51	1.50	5	150	8	2.83	1.31
6	175	6	3.72	1.85	6	175	4	3.47	1.39	6	175	9	2.78	1.26
7	200	7	3.22	1.30	7	200	4	3.34	1.21	7	200	10	2.73	1.20

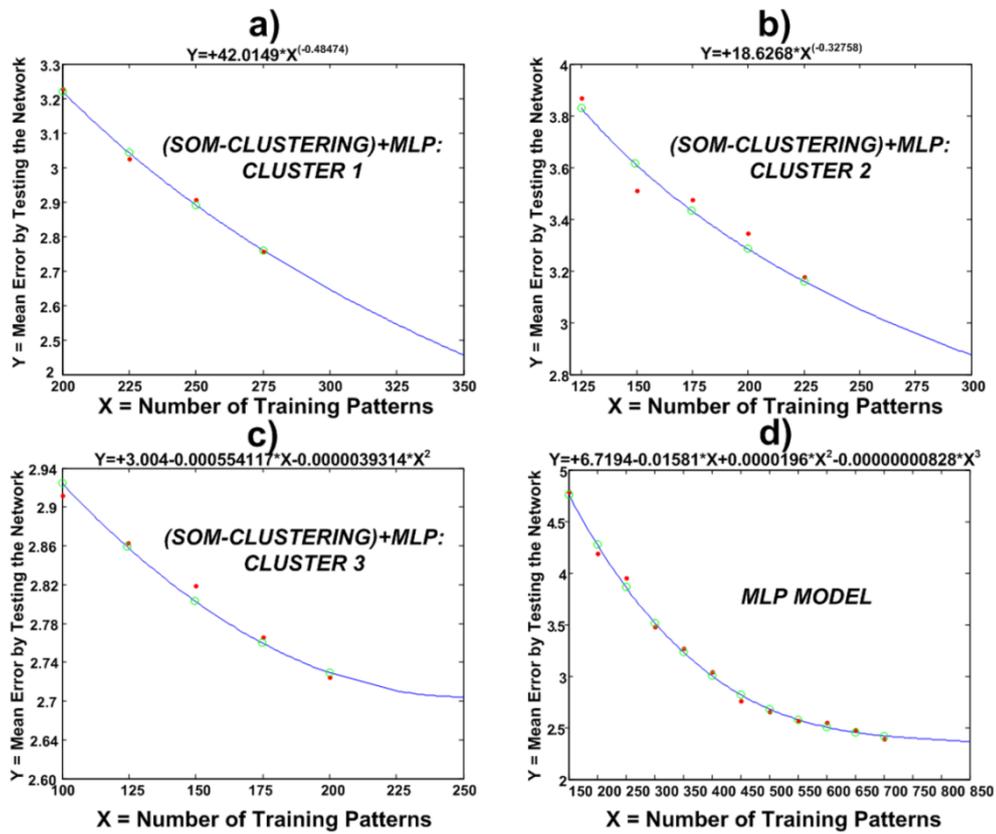


Figure 15. Evolution of the mean error in the operation phase versus the number of learning patterns: a) cluster 1 with 275 patterns; b) cluster 2 with 225 patterns; c) cluster 3 with 200 patterns; d) MLP trained with 700 patterns.