

Article

Improved Short-Term Load Forecasting Based on Two-Stage Predictions with Artificial Neural Networks in a Microgrid Environment

Luis Hernández ^{1,*}, Carlos Baladrón ², Javier M. Aguiar ², Lorena Calavia ², Belén Carro ², Antonio Sánchez-Esguevillas ², Javier Sanjuán ³, Álvaro González ⁴, Jaime Lloret ⁵

¹ CIEMAT (Research Centre for Energy, Environment and Technology), Autovía de Navarra A15, salida 56, 42290 Lobia, Soria, Spain

² Universidad de Valladolid, E.T.S.I. Telecomunicación, Campus Miguel Delibes, Paseo de Belén 15, 47011 Valladolid, Spain; E-Mails: cbalzor@ribera.tel.uva.es (C.B.); javagu@tel.uva.es (J.M.A.); lcaldom@ribera.tel.uva.es (L.C.); belcar@tel.uva.es (B.C.); antsan@tel.uva.es (A.S.-E.)

³ Universidad de Zaragoza, Escuela de Ingeniería y Arquitectura, 50018 Zaragoza, Spain; E-Mail: jsanju@unizar.es

⁴ Universidad de Zaragoza, Ingeniería Informática, 50018 Zaragoza, Spain; E-Mail: 554330@unizar.es

⁵ Departamento de Comunicaciones, Universidad Politécnica de Valencia, Camino Vera s/n, 46022 Valencia, Spain; E-Mail: jlloret@dcom.upv.es

* Author to whom correspondence should be addressed; E-Mail: luis.hernandez@ciemat.es; Tel.: +34-97-5281013; Fax: +34-97-5281051.

Received: 18 July 2013; in revised form: 18 August 2013 / Accepted: 23 August 2013 /

Published: 29 August 2013

Abstract: Short-Term Load Forecasting plays a significant role in energy generation planning, and is specially gaining momentum in the emerging Smart Grids environment, which usually presents highly disaggregated scenarios where detailed real-time information is available thanks to Communications and Information Technologies, as it happens for example in the case of *microgrids*. This paper presents a two stage prediction model based on an Artificial Neural Network in order to allow Short-Term Load Forecasting of the following day in *microgrid* environment, which first estimates peak and valley values of the demand curve of the day to be forecasted. Those, together with other variables, will make the second stage, forecast of the entire demand curve, more precise than a direct, single-stage forecast. The whole architecture of the model will be presented and the results compared with recent work on the same set of data, and on the same location, obtaining a Mean Absolute Percentage Error of 1.62% against the original 2.47% of the single stage model.

Keywords: artificial neural network; short-term load forecasting; microgrid; multilayer perceptron; peak load forecasting; valley load forecasting; next day's total load

1. Introduction

The term *Smart Grid (SG)* is associated with the new concept of “*smart*” electricity distribution networks, whose aim is the introduction of intelligence (through *Information and Communications Technologies—ICT*) for the optimization of the production and distribution of electricity.

In trying to meet the electricity demand with sufficient energy, utilities need to anticipate this demand by using estimate forecasts, usually 24 h ahead, and thus be able to know if they will need to buy energy in the market (energy defect), or sell it (excess energy). This is known as *Short-Term Load Forecasting (STLF)*, and helps in planning the operation of generators and energy related systems owned by the utility. Nowadays, research on areas such as *Demand Response (DR)* [1] and *Demand Dispatch (DD)* [2] is being conducted in order to involve *SG* and *STLF* techniques.

As shown in [3], *SG* enables the bi-directional flow of electric energy and information between utilities and consumers. It facilitates the integration within the network of the increasingly popular renewable generation sources, by promoting the participation of end users in energy saving and cooperating with the *DR* mechanism. The main objective of *DR* is the reduction of peak load within its control environment, whether it is a distribution network, a *Smart City (SC)* or a *microgrid*.

As shown in [4], valley periods require the utilities to reduce the production of generators with respect to peak periods. This production adaptation is especially difficult when renewable energies enter the equation, because their output power is much more difficult to forecast.

There are two groups that use *Peak Load Forecasting (PLF)*. Regarding to the first group, several models have been employed in the literature for *PLF* with different horizons. Regarding to the second group, other studies have employed *PLF* to facilitate *STLF*, since the estimation of this value becomes fundamental to improve the forecasting models.

Regarding *PLF*, there have been several works reported with different horizons. Mohan and Kumar [5] present a model based on an *Artificial Neural Network (ANN)* in order to make a *PLF* of the next day and up to seven days in advance. It achieves a range of *Mean Absolute Percentage Error (MAPE)* of 1.06% to 3.39% for the next day and of 1.72% to 4.15% for a forecasting horizon of 7 days.

With regard to long-term *PLF*, Hyndman and Fan [6] show that long-term demand forecasting plays an important role for the future of production plants. Authors advocate for a probabilistic approach to long-term peak load forecasting, and to support this model, they present a methodology to forecast the probability density of long-term peak load.

McSharry *et al.* [7] study the evolution of peak load along ten years, considering the impact of calendar variables such as day of the week and month. Then they present a model using weather variables, seasonality and calendar as input data.

There have been studies of *PLF* through hybrid systems, a combination of several prediction models in cascade to obtain the final estimation. The work in [8] presents a hybrid model for *PLF* of the next day, using a previous classification of similar patterns, and introducing climatic variables to improve the forecasting. To carry out the classification the *Self-Organizing Map (SOM)* model is used,

and then a specialized *ANN MultiLayer Perceptron (MLP)* is applied for *PLF* inside each cluster. Depending on the cluster used, *MAPEs* range from 1.56% to 3.51%.

Another hybrid system is presented in [9], which includes three different solutions for annual *PLF*. In the worst case a *MAPE* of 32% and in the best case of 3.7% are obtained. The work published in [10] shows a hybrid system for *PLF* of the next day. Authors propose a module for data decomposition into low and high frequency using Wavelets. The *MAPE* varies from 1.2% to 1.57%.

A disadvantage of previous studies (except [7]) is that they need lots of patterns for the learning phase. In particular, [5] needs three years of data, [6] uses 10 years, [8] employs four years, [9] needs over 15 years to obtain certain indexes that will be used in the estimation and [10] employs five years. As for [7], it only needs one year of data in the learning phase, but achieves a *MAPE* of 2.52%. The model proposed in this work, only requires two years of data, much lower values than at [5,6,8,9], and a similar values to the one used in [7], with the difference that the *MAPE* obtained with the model proposed here is 1.62%.

Some studies of the second group will be then presented. These works use *PLF* as input parameter of the model which will make *STLF*. Jain and Satish [11] present a hybrid system based on statistical models and *ANN* to make *STLF*. This system is composed by four modules and the system uses variable load and weather.

Amral *et al.* [12] present three different models to make *STLF*. The first model delivers 24 values at the same time using a *MLP*. The second model delivers peak load and valley load through two *MLP*, for later use them with statistical average of the load curve and to deliver the next day *STLF*. The third model features 24 *MLPs* in parallel.

The interesting feature of [11] lies in the synergy of models obtained from their combination, to get then a more accurate estimation through this combination. What is remarkable in [12] is the use of peak load and valley load to subsequently, by an average of load curves, to make the next day *STLF*. The model presented in this paper will be based on the potential of several stages to obtain *STLF* and, in addition, in the use of characteristic parameters of the load curve, such as: two peak loads, two valley loads and aggregate demand; all of them being estimates of the next day.

Everything presented so far corresponds to environments of vast areas with high populations (for example, in reference [5]: Haryana (India), with a population of 21 million people) and with high power consumption which will make demand habits overlap, relaxing the profiles of the load curve. The ranges of consumption of the cited publications are: 500–3500 MW [4]; 1000–3000 MW [6]; 1000–1800 MW [7]; 2500–4600 MW [8]; 1350–1800 MW [9]; 20,000–42,000 MW [10]; 9000–15,000 MW [11]; and 200–400 MW [12].

According to the *Consortium for Electric Reliability Technology Solutions (CERTS)* [13] a *microgrid* is: “(...) an aggregation of loads and microsources operating as a single system providing both power and heat. The majority of the microsources must be power electronic based to provide the required flexibility to insure operation as a single aggregated system. (...)”. It seems logical to think that for the proper functioning of a *microgrid*, it is essential to bear in mind demand behavior, requiring its forecasting and subsequently trying to adjust the *microgrid* generation to such claim. These environments require classification algorithms and clustering of the load curves to subsequently forecast the demand, as show in [14].

This paper presents a model based on *ANN* which makes *STFL* of the next day in a *microgrid* environment, using variables such as input to the network and estimated values of the day intended to be forecast. These points of interest are: the two maximum values (peak load), the two minimum ones (valley load) and aggregate demand. These variables together with other variables will make the forecast fit better to the model compared to another one with data of the same location. The objective will be to achieve accurate forecasting using information of the curve to forecast. The analyzed models used the estimation of peak load and valley load of the day to forecast, but not the two peaks and two valleys of the curve to forecast. The model will be compared with reference [15], where the location and data are the same.

There are two main differences between the method proposed along this work and other documented approaches: first, it is based on a two stage predictor, which forecasts a set of meaningful intermediate parameters (peaks and valleys in the load curve) as an input for the final prediction of the load curve; and second, it employs less data (only two years) than most of the methods reported in the literature. The paper is organized in the following way: Section 2 presents the methodological framework and data. Section 3 shows the structure of each of the employed *MLP*. Section 4 validates the model with real data. Section 5 analyzes the results. Section 6 presents the conclusions.

2. Data Description and Methodology Framework

2.1. Load Data

Iberdrola provided a historic data set from 1 January 2008 until 31 December 2010, of the capital of the province of Soria (Castilla y León, Spain), which presents one sample per hour. The consumption, characteristic of a *microgrid*, varies between 7–39 MW, and does not conform to the pattern of values of a country or a wide region. After deleting the existing erroneous patterns of the 3 years of available data, 70% have been used for learning (70% training ratio, 15% validation ratio and 15% test ratio) of the *MLP*, and the remaining 30% for the validation phase.

The meteorological data used in this study were collected by the *Spanish Meteorological Agency AEMET* from the meteorological station installed in Soria. The meteorological data were collected from 1 January 2008 to 12 October 2010. The weather variables considered are: precipitation (mm), air temperature (°C), average wind speed (m/s), average wind direction (sexagesimal degrees), relative humidity (%), pressure (hPa) and global solar radiation.

Each row presents data about date and hour of registration, minute of registration, source meteorological station, altitude, name of the province, longitude, latitude, precipitation, ambient temperature, average wind speed, average wind direction, relative humidity, pressure and global solar radiation. Weather variables—excluding global solar radiation, which is sampled hourly between 5:00 and 20:00—are monitored on a ten-minute basis. Missing values due to monitoring failures are recovered using interpolation (this is done to implement a system capable of working in real time). Then, air temperature, average wind speed, average wind direction, relative humidity and pressure are averaged in groups of six ten-minute intervals to obtain hourly measures. Precipitations are not averaged, but accumulated instead. Then, daily average values are calculated for all of the variables.

Microgrids must be capable of managing electric energy consumptions ranging between ten of kW to hundreds of MW. These consumptions are characteristic of small and medium sized cities, towns or

even smaller environments. Behavioral habits of these locations will make the load curves show the uneven and rough environments each hour, in contrast to those shown in more aggregated environments (large areas or countries with high density and very concentrated population and high electricity consumption) because in those areas the superposition of habits, and therefore the load curves, soften the final load curve.

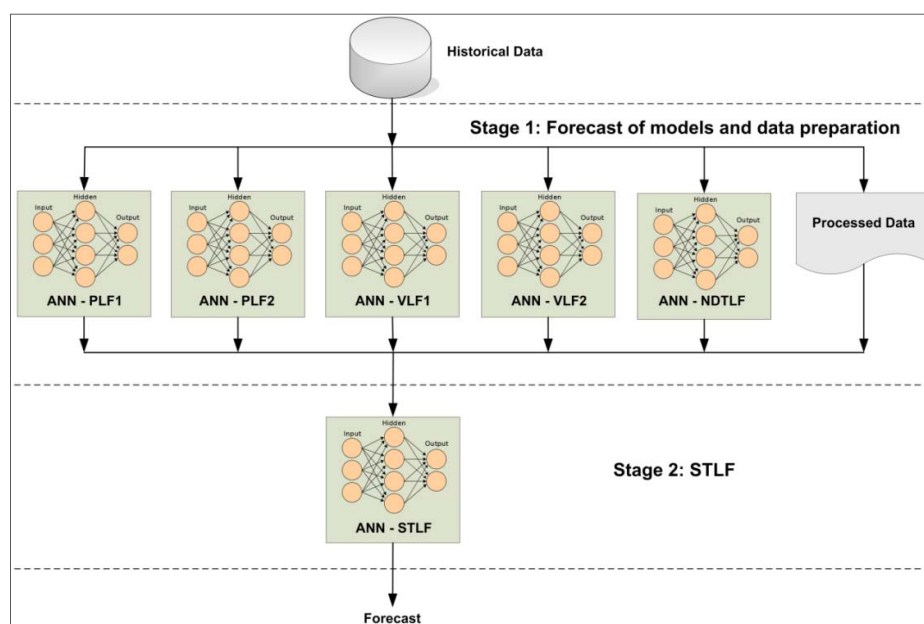
The hypothesis of this paper is that load curves have a very specific topology that contains a series of concepts (daily aggregate demand, two peak loads and two valley loads) that would be very important to know *a priori* because it would greatly facilitate the work of the predictor and improve their results. These peaks and valleys of the next day, obviously not available *a priori*, can be forecasted in a first stage (using predictors specialized in this type of forecasting), and then supply these forecasts to a second stage of forecasting, which will be responsible for predicting the load curve of the next day (*STLF*).

In this way, although the input data are the same in the case of the proposed two-stage predictor that in a direct predictor [15], the additional variables and the ability of the first stage (forecasting of peaks, forecasting of valleys and forecasting of daily aggregate demand) will improve the performance of the global system.

2.2. Methodology Framework

This paper presents a methodology for *STLF* through input variables based on estimated details of the forecasting day. The methodological framework (Figure 1) can be summarized in two stages: forecast of models and data preparation; and *STLFs* of the next day.

Figure 1. Block diagram of the proposed methodology.



All data mentioned in Section 2.1 are stored in a database, which will be connected with the first stage of the submitted methodological framework. Six different actions will be carried out at this early stage:

- *ANN-PLF1*: *MLP* network, it is responsible for the estimation of the first peak load of the day to forecast. This network will receive the input variables of the database and will present its estimation to the second stage;
- *ANN-PLF2*: *MLP* network, it is responsible for the estimation of the second peak load of the day to forecast. This network will receive the input variables of the database and will present its estimation to the second stage;
- *ANN-VLF1*: *MLP* network, it is responsible for the estimation of the first valley load of the day to forecast. This network will receive the input variables of the database and will present its estimation to the second stage;
- *ANN-VLF2*: *MLP* network, it is responsible for the estimation of the second valley load of the day to forecast. This network will receive the input variables of the database and will present its estimation to the second stage;
- *ANN-NDTLF*: *MLP* network, it is responsible for the estimation *Next Day's Total Load (NDTL)*. This network will receive the input variables of the database and will present its estimation to the second stage;
- Processed data: it is responsible for processing the data from the database expected as input variables by the *STLF* of the second stage.

The second stage will consist in an *MLP* network to make the next day *STLF (ANN-STLF)*. The network will need data from the first stage to be used as input variables. Once submitted these variables, the network will deliver the estimation of the 24 values of electric energy demand of the next day.

3. ANN Structure and Evaluating the Performance of ANN

Below each of the networks used in Section 2.2 is explained. Each of the layers (input, output and hidden) will be presented, and the variables of the input layer, the number of neurons in the hidden layer and the variables of the output layer indicated.

The way to optimize the *MLPs*—both to determine the number of neurons in the hidden layer and to establish the best learning algorithm—is usually performed by a heuristic method. Therefore, an automatic script developed in MatLab will be used, where all possible parameters (number of neurons in the hidden layer, learning function, network performance function during learning, *etc.*) will be varied to select the best topology for each of the proposed networks. For each of the models, for each of the training functions, and for each hidden layer size (between one and 20 neurons), the script executed 100 different runs in order to achieve statistically meaningful results which rule out the random factors influencing the *ANN* (such as the initial state). The training functions considered are: *traingd* is gradient descent backpropagation; *traingdm* is gradient descent with momentum backpropagation; *traingda* is gradient descent with adaptive learning rate backpropagation; *traingdx* is gradient descent with momentum and adaptive learning rate backpropagation; *trainrp* is resilient backpropagation; *traingcf* is conjugate gradient backpropagation with Fletcher-Reeves updates; *traingcp* is conjugate gradient backpropagation with Fletcher-Ribière updates; *traingcb* is conjugate gradient backpropagation with Powell-Beale restarts; *traingcg* is scaled conjugated gradient backpropagation;

trainbfg is BFGS quasi-Newton backpropagation; trainoss is one-step secant backpropagation; trainlm is Levenberg-Marquardt backpropagation; and trainbr is Bayesian regulation backpropagation.

The number of neurons in the hidden layer is detailed in the following sections of this paper, while the rest of parameters for all networks are common: learning function trainbr (Bayesian Regulation Backpropagation) and network performance function *Sum Squared Error (SSE)*.

Periodic/cyclical variables, such as day of the week and month (which are essential for the *ANN* to properly detect weekly, monthly and seasonal patterns), are supplied to the networks in the form of values of sines and cosines, as it has been demonstrated [16,17] that this transformation significantly improves the performance of the *ANN*.

3.1. ANN-PLFx Structure

The presented structure serves both for *ANN-PLF1* and *ANN-PLF2*, the only difference is that the network will use values from the first peak load when it is 1 or from the second peak load when it is 2. As reference of the network, results of [18] have been used, showing a network for *PLF* with a *MAPE* of 1.57% and two hidden layers. It has been taken into account [11] for selecting additional input variables.

Input: 9

- $PLi_{(d-1)}, PLi_{(d-2)}, PLi_{(d-3)}, PLi_{(d-4)}, PLi_{(d-5)}, PLi_{(d-6)}, PLi_{(d-7)}$: peak load of the 7 previous days of the day to forecast; i is 1 or 2;
- $PLT_{(d-1)}$: peak load temperature of the previous day of the day to forecast;
- $AvgT_{(d-1)}$: mean temperature of the previous day to forecast.

Output: 1

- PLi_d : peak load 1 or 2 of the day to forecast; i is 1 or 2.

Hidden: three neurons.

3.2. ANN-VLFx Structure

The presented structure serves both for *ANN-VLF1* and *ANN-VLF2*, the only difference is that the network will use values from the first peak load when it is 1 or from the second peak load when it is 2. The same philosophy that *ANN-PLF* is followed to make the network design.

Input: 9

- $VLi_{(d-1)}, VLi_{(d-2)}, VLi_{(d-3)}, VLi_{(d-4)}, VLi_{(d-5)}, VLi_{(d-6)}, VLi_{(d-7)}$: valley load of the 7 previous days of the day to forecast; i is 1 or 2;
- $VLT_{(d-1)}$: valley load temperature of the previous day of the day to forecast;
- $AvgT_{(d-1)}$: mean temperature of the previous day to forecast.

Output: 1

- VLi_d : valley load 1 or 2 of the day to forecast; i is 1 or 2.

Hidden: three neurons.

3.3. ANN–NDTLF Structure

Below it is shown the ANN–NDTLF network structure, indicating each of the variables used, both for the input layer and the output layer. Results obtained from [19] have been used as a reference of the network. In [19], after an analysis of the influence of climatic variables on aggregate demand, and the influence of aggregate demand regarding herself in previous days, multiple networks with different inputs are analyzed. The network with the best results is *Forecast with Aggregated Load-Working/non-working day-Day of the Week-Solar Radiation (FALWDWSR)*, with a MAPE of 2.98%.

Input: 24

- $TL_{(d-1)}, TL_{(d-7)}, TL_{(d-14)}, TL_{(d-21)}$: the total load of a day is clearly linked with the total load of the previous day and total loads of the same day of the week of the three previous weeks, regardless of the type of day, in terms of working/non-working day and day of the week. For this reason, network inputs of the previous day and of the three similar days regarding the day of the week of the three previous weeks, have been selected as total loads;
- $W_{(d-1)}, W_{(d-7)}, W_{(d-14)}, W_{(d-21)}, W_d$: working/non-working day ($holiday = 1$ and $working-day = 2$) of the days mentioned in the previous paragraph, as well as the working/non-working day of the day to forecast. The coding is ($holiday = 1$ and $working-day = 2$);
- $DW_{(d-1)}, DW_{(d-7)}, DW_{(d-14)}, DW_{(d-21)}, DW_d$: day of the week in sine and cosine form, both of the last days mentioned in the first point, as of the day to forecast. The coding is ($Sunday = 0$, $Monday = 1, \dots, Friday = 5, Saturday = 6$);
- $SW_{(d-1)}, SW_{(d-7)}, SW_{(d-14)}, SW_{(d-21)}, SW_d$: solar radiation of the last days referred to in the first point, as well as the day to forecast.

Output: 1

- $NDTL_d$.

Hidden: four neurons.

3.4. ANN–STLF Structure

Below it is shown the ANN–STLF structure, indicating each of the used variables, both for the input layer and the output layer.

Input: 33

- $L_{(d-1)1}, L_{(d-1)2}, L_{(d-1)3}, \dots, L_{(d-1)24}$: corresponding to the 24 values of the load curve of the previous day of the day to forecast;
- Day of the week $d-1$: this variable is introduced as two, in sines and cosines form, through $\sin[(2 \cdot \pi \cdot day) / 7]_{(d-1)}$ and $\cos[(2 \cdot \pi \cdot day) / 7]_{(d-1)}$, with day values from 0 to 6 ($Sunday = 0$, $Monday = 1$, $Tuesday = 2$, $Wednesday = 3$, $Thursday = 4$, $Friday = 5$, $Saturday = 6$);
- Month $d-1$: this variable is introduced as two, in sines and cosines form, through $\sin[(2 \cdot \pi \cdot month) / 12]_{(d-1)}$ and $\cos[(2 \cdot \pi \cdot month) / 12]_{(d-1)}$, with $month$ values from 1 to 12 ($January = 1$, $February = 2$, $March = 3, \dots, December = 12$);
- $PL1_d$ y $PL2_d$: two maximum values of the load curve of the day to forecast (peak load 1 and peak load 2);

- $VL1_d$ y $VL2_d$: two minimum values of the load curve of the day to forecast (valley load 1 and valley load 2);
- $NDTL_d$.

Output: 24

- $L_{(d)1}, L_{(d)2}, L_{(d)3}, \dots, L_{(d)24}$: corresponding to the 24 values of the load charge of the day to forecast.

Hidden: 22 neurons.

3.5. Evaluating the Performance of ANN

For all the validation phase, in which the day load curve forecasting will be obtained, $MAPE$ is estimated by means of:

$$MAPE_d = 100 \cdot \frac{\sum_{i=1}^{24} \left| \frac{L_{(d)i} - \hat{L}_{(d)i}}{L_{(d)i}} \right|}{24} \quad (1)$$

where $L_{(d)i}$ represents the measured value; $\hat{L}_d(i)$ represents the estimated value and i the hour in both cases.

Once $MAPE_d$ has been obtained, the mean error along all samples in the validation phase is estimated by means of:

$$ERROR_{OP} = \frac{\sum_{i=1}^n MAPE_{di}}{n} \quad (2)$$

where n represents the number of days of the validation phase.

To examine how the forecasting error is reflected on the load curve, error is displayed on a graphic including all forecasted days in the validation phase. Using this method, the forecast mean error for each of the 24 h is obtained by means of:

$$MAPE_h_i = 100 \cdot \frac{\sum_{k=1}^n MAPE_{i,k}}{n} \quad (3)$$

with $i = 1, 2, \dots, 24$; n stands for the sample size in the validation phase and $MAPE_{i,k}$ the hourly error i for the day k .

Although the aim of this paper is ANN-STLF, the mean error of each of the networks belonging to the first stage will be delivered by means of Equation (1). Subsequently, the results will be grouped by months [using Equation (2)] and for each network.

4. Validation Results

4.1. Results

Table 1 shows $MAPE$ gathered by months for each network of the first stage. $MAPEs$ obtained for PLF and VLF are optimal if they are compared with the results of [5], which $MAPE$ varies between

1.06% and 3.39%, and with the ones of [6], which *MAPEs* for the five clusters are: 1.56%, 2.13%, 3.51%, 2.04% and 2.01%. On the other hand, only *ANN-VLF2* could approximate the results of [10], with *MAPE* between 1.2% and 1.57%.

Table 1. Daily *MAPE* of the validation phase. *MAPE* is grouped by months. All the networks of the first stage are represented.

Month	<i>MAPE</i> (%)	<i>MAPE</i> (%)	<i>MAPE</i> (%)	<i>MAPE</i> (%)	<i>MAPE</i> (%)
	<i>ANN-PLF1</i>	<i>ANN-PLF2</i>	<i>ANN-VLF1</i>	<i>ANN-VLF2</i>	<i>ANN-NDTLF</i>
February	2.03	2.04	1.92	1.88	2.39
March	2.35	2.38	2.02	1.96	2.71
April	2.30	2.15	2.12	2.05	3.26
May	2.13	2.16	1.96	1.95	2.47
June	2.30	2.22	2.09	2.00	4.50
July	2.04	2.08	1.87	1.90	2.63
August	2.13	2.09	2.01	1.99	2.74
September	2.20	2.15	2.03	2.04	1.20
October	2.84	2.67	2.22	2.10	4.86
November	2.24	2.20	2.05	2.00	4.19
December	2.99	2.53	2.17	2.02	3.31
Annual average	2.32	2.24	2.04	1.99	3.11

Once the validation phase finishes (*ANN-STLF*), the evolution of (1) is represented in Figure 2 for each of the days of this stage. The mean error of all the validation phase is given by (2), with a value of 1.62%. In the same figure, *MAPE* is represented together with: the mean, mean \pm standard deviation and mean $\pm 2 \times$ standard deviation. The figure uses a specific nomenclature with the format: “2 1/12-1 1”, and it can be interpreted as follows: “2” represents working/non-working day of the previous day; “1” corresponds to the month of the day to forecast (January); “12” represents the day of the month to forecast; “1” designates the day of the week (Monday) of the day forecasting; and “1” is working/non-working day of the day forecasting.

In [12] three models for *STLF* (next day) were analyzed. In particular, the second model used *PLF* and *VLF* (maximum and minimum points respectively) as data of a statistical estimator that obtained *STLF*, with a *MAPE* of 2.81%. *MAPE* presented herein substantially improves the second model.

In addition, reference [15] proposes a network with different input variables, but with data in the same location and the same period of time, even though [15] uses the same set of data for the learning and validation phases. *MAPE* obtained in [15] is 2.47%. This value is greater than the 1.62% obtained in the present work. Thus, using estimated values of the day to forecast clearly improves the results of the operating phase. In particular, this improvement is obtained with the information of the two peaks and the two valleys of the load curve to forecast.

Figure 2. Errors per day in validation phase. In the *x-axis* are the days. In the *y-axis* are the errors by Equation (2).

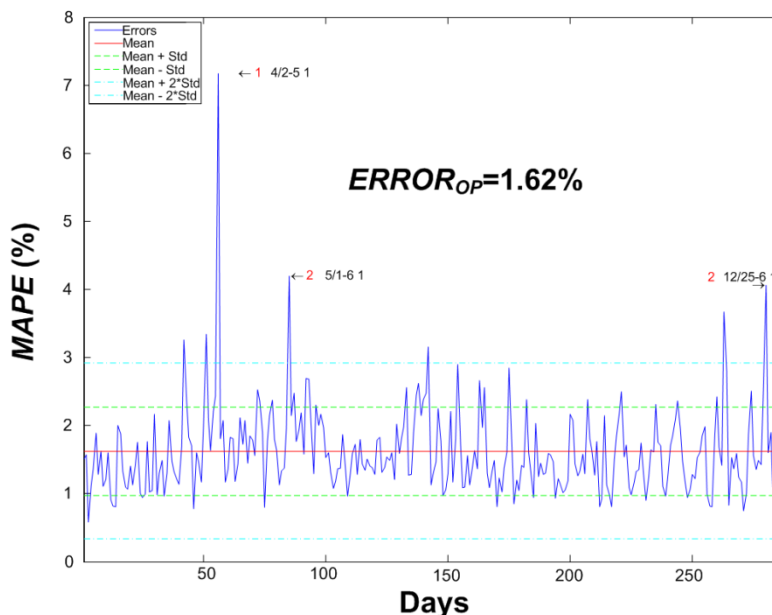
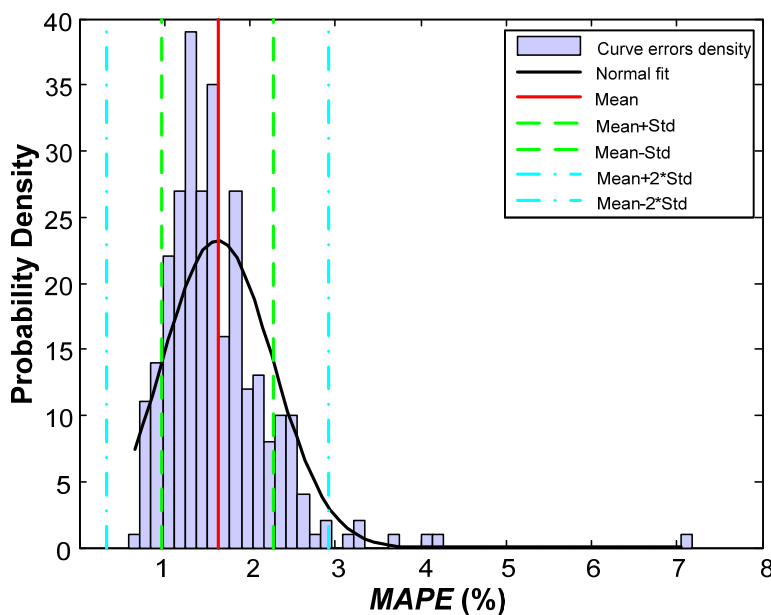


Figure 3 shows the errors as *Probability Density Function (PDF)*, where intervals between the mean, mean \pm standard deviation and mean $\pm 2 \times$ standard deviation are represented. As shown in the figure, most of the errors are in the first interval (79%). The percentage of distribution of errors is shown in Table 2.

Figure 3. Curve errors data in validation phase. In the *x-axis* are errors by Equation (1). In the *y-axis* are the probability densities.



In [15], a standard deviation of 0.95% was obtained while 0.65% has been obtained in this study. It shows that errors are concentrated around the mean error, which is lower than in the other study. In this work the 79% of the daily errors are inside the mean standard deviation, however in [15] this

percentage fell to 72%. The percentage of daily errors over the mean standard deviation ($\pm 2\times$) in this study is 2.45%, lower than the 4.21% obtained at [15], so very high errors have decreased. Table 2 shows these results.

Table 2. Error Distribution per day.

Variable	Value	Percentage (%)
Mean	0.162	1.62
Standard deviation (Std.)	0.0065	0.65
No. of errors above $\times 1$ Std.	34	11.89
No. of errors between $\times 1$ Std.	227	79.37
No. of errors below $\times 1$ Std.	25	8.74
No. of errors above $\times 2$ Std.	7	2.45
No. of errors between $\times 2$ Std.	279	97.55
No. of errors below $\times 2$ Std.	0	0.00

Figure 4 shows the errors per hour of the operating phase [using Equation (3)]. As shown in [15], the forecast continues to have its highest errors in the vicinity of the singular points already mentioned (peaks and valleys), which coincide with the changes in turning points that occur near there.

Figure 4. Errors per hour in validation phase. In the *x-axis* are the 24 h. In the *y-axis* are the errors by Equation (3).

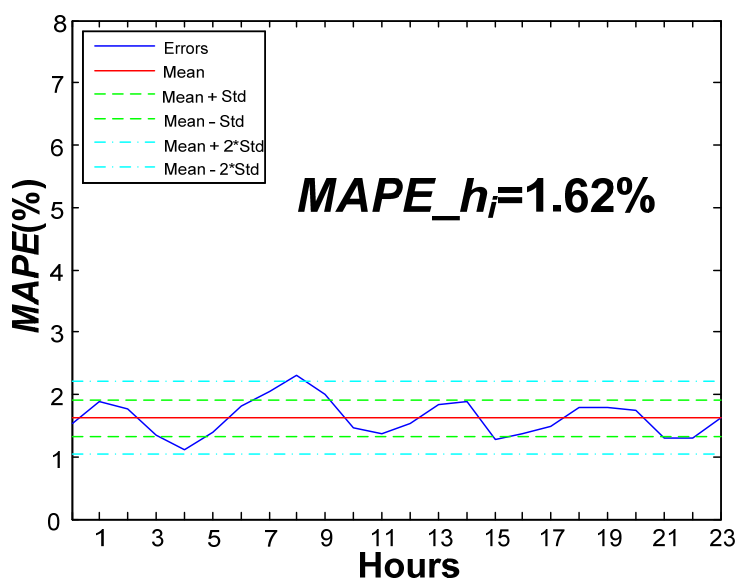


Figure 5 shows the errors per hour as a *PDF*, where intervals between the mean, mean \pm standard deviation and mean $\pm 2\times$ standard deviation are represented. As shown in the figure, most of the errors are in the first interval (70%).

The percentage of distribution of errors is shown in Table 3. In this work, 79% of the errors per hour are inside the average and mean deviation, while in reference [15] the percentage was 62.5%, so the errors per hour are more concentrated around the average.

Figure 5. Curve errors per hour in validation phase. Errors by Equation (3) are represented in the *x*-axis. In the *y*-axis are the probability densities.

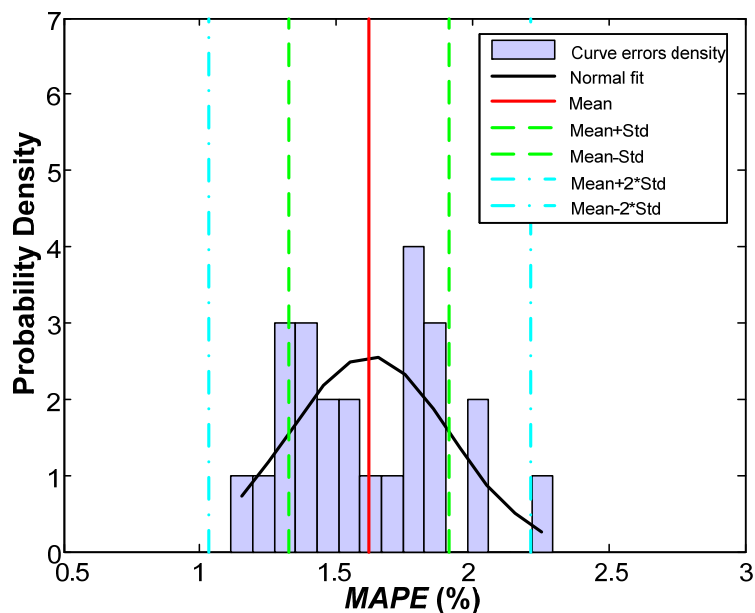
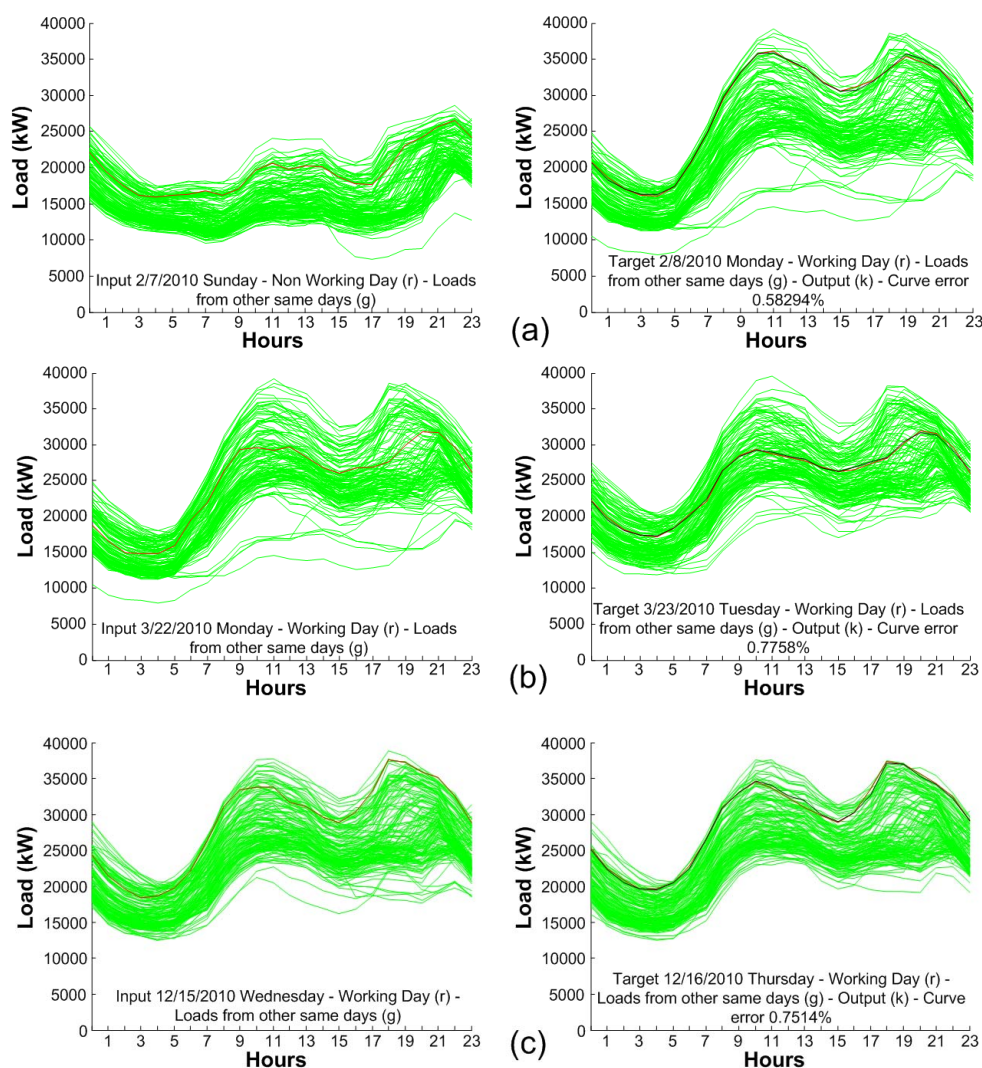


Table 3. Error Distribution per hour.

Variable	Value	Percentage (%)
Mean	0.0162	1.62
Standard deviation (Std.)	0.0065	0.65
No. of errors above \times 1 Std.	3	12.50
No. of errors between \times 1 Std.	17	70.83
No. of errors below \times 1 Std.	4	16.67
No. of errors above \times 2 Std.	1	4.17
No. of errors between \times 2 Std.	23	95.83
No. of errors below \times 2 Std.	0	0.00

Figure 6 shows forecasts for the three days with the lowest mean error: (a) represents the forecasting of 02/02/2010 with an error of 0.58%; (b) represents the forecasting of 03/23/2010 with an error of 0.77%; (c) represents the forecasting of 12/16/2010 with an error of 0.75%.

The lowest errors of reference [15] were 1.20%, 1.10% and 1.13%, all higher than the results of the present study. In relation to Figure 6, the graph on the left shows in red the curve of the previous day of the day to forecast and in green all the curves that have the same characteristics, regarding the day of the week and the working/non-working day. The graph on the right shows in red the real curve which is intended to forecast, in black the forecast and in green all the curves with the same characteristics that the day to forecast, regarding the day of the week and the working/non-working day.

Figure 6. Forecasts with low mean error: (a) 02/08/2010; (b) 03/23/2010; (c) 12/16/2010.

4.2. Computational Cost

MatLab was employed to implement the *ANN* and the rest of the scripts developed for additional tasks (error estimation, figures, *etc.*). A desktop computer with an Intel Core2 vPro 3.4 GHz 2GB RAM processor was used.

Once the different networks were optimized, and for all of them, both the learning phase (70% of data) and validation phase (30% of data) have been carried out 100 times. Table 4 shows the measurement of time for each of the two phases and the network.

By observing the results we can conclude that in real operation, the time necessary to obtain the 24 values of demand for the next day will be about 1 second. It should take into account, that for getting a forecast, the estimations of the four networks in the first stage could be made in parallel, so the time to dispose of the five values in the worst case would be 0.38 s, and this should be added to the time needed by *ANN-STLF* (0.62 s). It is assumed that Processed Data has an execution time of less than 0.38 s.

Table 4. Time used in the learning phase and validation phase for each network.

Forecasts	Learning phase time	Validation phase time
<i>ANN-PLF1</i>	8 min and 17 s	57 s
<i>ANN-PLF2</i>	8 min and 25 s	58 s
<i>ANN-VLF1</i>	8 min and 02 s	53 s
<i>ANN-VLF1</i>	8 min and 05 s	54 s
<i>ANN-NDTLF</i>	15 min and 45 s	1 min and 50 s
<i>ANN-STLF</i>	22 min and 15 s	2 min and 57 s

In [5], *PLF* with *MLP* trained with *Scaled Conjugate Gradient Back Propagation (SCGBP)* spends 104.79 s in the learning phase for a similar data set. The time is higher than that achieved in this work for *PLF*. As shown in [10], for a double training phase and *PLF*, including the learning of the genetic algorithm, the validation and the *ANN* computation spends 705 s in the worst case and 163 s for the model needing less time.

5. Result Analysis

5.1. Error Distribution

As shown in Figure 2, and supported by the data reflected in Figure 3 and Table 2, the mean error of the entire validation phase [given by Equation (3)] is understood to be within 1.55% and 1.68%. These results are better than those of reference [15], that ranged between 1.45% and 3.35%. By observing the hourly mean error shown in Figure 4, and the data displayed in Figure 5 and Table 3, it is possible to see that, although better results than those of [15] have been obtained, the most significant errors occur at the characteristic points of the forecast load curve. Therefore, the contribution of information (four characteristic points and the total demand) improves the forecasting substantially with respect to [15]. However, it is still interesting to provide new information to the network to improve results. This information should focus on those areas of the curve.

5.2. Errors per Day of the Week and Month

Table 5 shows the daily mean error of the validation phase per day of the week. The results have been compared with those obtained in [15]. The error is reduced significantly, and it is necessary to emphasize the decrease of 1.03% on Sundays.

Table 6 represents the monthly mean error of validation phase. Results have been compared with those obtained in [15]. Getting the difference between *MAPEs* for each month, and gathering them into warm months (April, May, June, July, August and September), and cold months (October, November, December, February and March) the average of the differences between these groups are made. *MAPE* is decreased 0.96% and 0.65% for each group, respectively, in favor of the network. Cold months have greater turning points, so the contribution of information of these points improves forecasting more than for the other months where the differences are not so obvious.

Table 5. Daily *MAPE* (%) of validation phase. Comparison of *ANN-STLF* with results obtained in [15].

Model	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
[15]	2.92	2.31	2.01	1.97	2.38	2.60	2.59
<i>ANN-STLF</i>	1.89	1.57	1.34	1.41	1.56	1.64	1.91

Table 6. Monthly *MAPE* (%) of validation phase. Comparison of *ANN-STLF* with results obtained in [15].

Model	February	March	April	May	June	July	August	September	October	November	December
[15]	2.11	2.68	2.52	2.25	2.52	2.16	2.18	2.27	3.07	1.98	2.69
<i>ANN-STLF</i>	1.32	1.59	1.93	1.81	1.78	1.53	1.45	1.48	1.72	1.41	1.66

5.3. Error Analysis

Hernández *et al.* [15] explain why there were high errors in the forecasting of certain days when using the network of that work. The main reasons were: few pairs of patterns in the learning phase similar to the day with high error in their forecasting; forecasting on holiday Saturdays, since the training patterns did not present this characteristic; days on weekdays (not holidays) immersed in holidays of the locality (Jueves la Saca); day before of holiday weekdays with an unusually low energy consumption; Christmas Eve with low energy consumption.

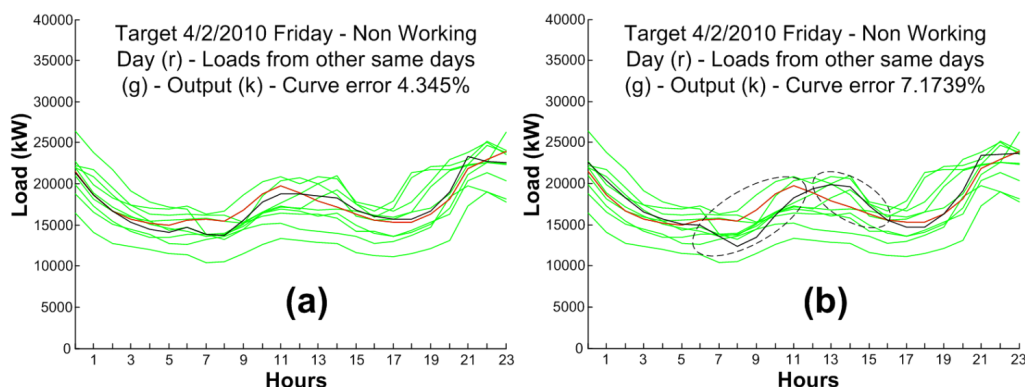
Table 7 presents the results of the days with high error in [15] (along with a few others), to show that the structure of input variables of this study improves significantly these forecasts, except for 04/02/2010. It is important to highlight the striking reduction for 12/25/2010, 12/26/2010 and 12/31/2010.

Figure 7 represents the forecasting of 04/02/2010, using [15] in Figure 7a and the approach here presented in Figure 7b. The real curve (red) ascends prematurely around 5 h and then declines. Finally it ascends after 7–8 h. In this case, *VLF1* seems to be the cause of the final error of *ANN-STLF* because in option Figure 7b of the black curve (forecasting), the peak appears delayed on the forecasting and makes both the forecasting of the first ascent and of the second descent (circles in the figure) accumulate a high error.

Finally, the forecasts with lowest errors are listed. They match up with all working days from Monday to Friday. It is important to remember that these days have the highest number of patterns. The following days have a *MAPE* lower than 0.92%: 02/02/2010—0.58%; 02/17/2010—0.91%; 02/18/2010—0.81%; 02/19/2010—0.81%; 03/23/2010—0.77%; 04/21/2010—0.79%; 07/27/2010—0.80%; 08/03/2010—0.84%; 09/09/2010—0.80%; 09/10/2010—0.92%; 09/14/2010—0.81%; 09/28/2010—0.90%; 12/02/2010—0.81%; 12/03/2010—0.80%; 12/10/2010—0.83%; 12/16/2010—0.75%; 12/28/2010—0.89%.

Table 7. Daily *MAPE* (%) of [15] with higher error and a comparison with *ANN-STLF*. Days are shown as MM/DD because all of them belong to 2010.

Model	04/02	05/01	06/24	06/25	06/27	06/29	07/11	10/11	12/06	12/08	12/25	12/26	12/31
[15]	4.34	4.77	3.92	3.84	4.04	4.36	4.07	4.54	4.77	4.73	8.04	5.06	4.8
<i>ANN-STLF</i>	7.17	4.19	2.44	2.62	2.38	3.15	2.89	2.35	1.68	3.66	4.05	1.59	1.53

Figure 7. Forecast for 04/02/2010 (a) [15]; (b) ANN-STLF.

6. Conclusions

Imagine a *microgrid* that has a base generation plant (hydraulic microturbine); in order to carry out peak shaving, knowledge of the peak load will be necessary so as to know the amount of water to be used the next day. As another example, *Virtual Power Plants (VPPs)* pose a challenge for demand forecasting and generation, and a possible approach is to use a management model which takes into account the multiple elements that are part of it, making them cooperate to obtain a demand forecast via *ANN* in disaggregated environments, as shown by Hernández *et al.* [20].

The main contribution of this paper is demonstrating that for each particular load forecast scenario a specific analysis of the influencing factors and a two stage predictor (to forecast those intermediate factors) has the potential to give better results than directly forecasting the variables of interest. In the use case presented along this work, an intermediate estimation of peak and valley values offers results that are 52% more accurate than a direct, one-stage prediction of the next day hourly load curve.

Furthermore, the test has been made with disaggregated environment data, where the zones near to the characteristic points are rougher than in more aggregated environments (large areas or countries). The deployment of *microgrids* is an imminent issue, so it will require its control and operation. As for consumption, a disaggregation of the load curve occurs, complicating the forecasting. In addition, there is a difficulty in demand forecasting. In addition, *microgrid* environments are ideal to make *DR*. *DR* is not conceived without having the most exact possible knowledge of the demand; therefore, the development of models based on *ANN* for *STLF* will be necessary, with much room for development and improvement.

With the model here presented, the forecast errors obtained for peaks and valleys are: 2.03%–2.99% (*PLF1*), 2.04%–2.67% (*PLF2*), 1.92%–2.22% (*VLF1*), 1.88%–2.10% (*VLF2*) and 1.20%–4.86% (*NDTLF*). The results of valley load estimation are good and they are compared with the figures reported in [6] (1.56%–3.51%). Valley estimation shows better results than peak estimation, mainly due to the fact that peak height and location has a greater variance than valleys. The *MAPE* of the validation phase is 1.62%, substantially better than the one obtained in previous works (2.47%) for the same set of data. A raw improvement of 0.85% in *MAPE* is a great advance, representing an overall relative improvement of 52%.

Acknowledgments

We would like to thank the Spanish National Meteorology Agency, Department of Environmental Affairs (AEMET) for the meteorological information they so kindly provided to us, and Iberdrola, especially to Óscar Villanueva, Head of Business Exploitation in Burgos and Soria, and Silvia Herrero, Head of Communication of Iberdrola in Castilla y León for the power demand information provided. We would also like to thank Laura Sanjuán and Marta Serna for the translation and revision of this paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Simmhan, Y.; Aman, S.; Cao, B.; Giakkaupis, M.; Kumbhare, A.; Zhou, Q.; Paul, D.; Fern, C.; Sharma, A.; Prasanna, V. *An Informatics Approach to Demand Response Optimization in Smart Grids*; Technical Report for University of Southern California: Los Angeles, CA, USA, 2011.
2. Brooks, A.; Lu, E.; Reicher, D.; Spirakis, C.; Weihl, B. Demand dispatch. *IEEE Power Energy Mag.* **2010**, *8*, 20–29.
3. Chan, S.C.; Tsui, K.M.; Wu, H.C.; Hou, Y.; Wu, Y.-C.; Wu, F.F. Load/price forecasting and managing demand response for smart grids. *IEEE Signal Process. Mag.* **2012**, *29*, 68–85.
4. O’Toole, E.; Clarke, S. Dynamic Forecasting and Adaptation for Demand Optimization in the Smart Grid. In Proceedings of 2012 International Workshop on Software Engineering for the Smart Grid (SE4SG), Zurich, Switzerland, 3 June 2012; pp. 30–33.
5. Saini, L.M.; Soni, M.K. Artificial neural network-based peak load forecasting using conjugate gradient methods. *IEEE Trans. Power Syst.* **2002**, *17*, 907–912.
6. Hyndman, R.J.; Fan, S. Density forecasting for long-term peak electricity demand. *IEEE Trans. Power Syst.* **2010**, *25*, 1142–1153.
7. McSharry, P.E.; Bowman, S.; Bloemhof, G. Probabilistic forecasts of the magnitude and timing of peak electricity demand. *IEEE Trans. Power Syst.* **2005**, *20*, 1166–1172.
8. Amin-Naseri, M.R.; Soroush, A.R. Combined use of unsupervised and supervised learning for daily peak load forecasting. *Energy Convers. Manag.* **2008**, *49*, 1302–1308.
9. Maksimovich, S.M.; Shiljkut, V.M. The peak load forecasting afterwards its intensive reduction. *IEEE Power Deliv.* **2009**, *24*, 1552–1559.
10. Moazzami, M.; Khodabakhshian, A.; Hooshmand, R. A new hybrid day-ahead peak load forecasting method for Iran’s National Grid. *Appl. Energy* **2013**, *101*, 489–501.
11. Jain, A.; Satish, B. Integrated Approach for Short-Term Load Forecasting Using SVM and ANN. In Proceedings of IEEE TENCON 2008 Region Conference, Hyderabad, India, 19–21 November 2008; pp. 1–6.
12. Amral, N.; King, D.; Ozveren, C.S. Application of Artificial Neural Network for Short Term Load Forecasting. In Proceedings of 43rd International Universities Power Engineering Conference (UPEC), Padova, Italy, 1–4 September 2008; pp. 1–5.

13. Lasseter, R.; Akhil, A.; Marnay, C.; Stevens, J.; Dagle, J.; Guttromson, R.; Meliopoulos, A.S.; Yinger, R.; Eto, J. *White Paper on Integration of Distributed Energy Resources. The CERTS Microgrid Concept*; Consortium for Electric Reliability Technology Solutions (CERTS): Los Angeles, CA, USA, 2002.
14. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A. Classification and clustering of electricity demand patterns in industrial parks. *Energies* **2012**, *5*, 5215–5228.
15. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A.; Lloret, J. Short-term load forecasting for microgrids based on artificial neural networks. *Energies* **2013**, *6*, 1385–1408.
16. Drezga, I.; Rahman, S. Phase-Space Short-Term Load Forecasting for Deregulated Electric Power Industry. In Proceedings of the International Joint Conference on Neural Networks, Washington, DC, USA, 10–16 July 1999; Volume 5, pp. 3405–3409.
17. Razavi, S.; Tolson, B.A. A new formulation for feedforward neural networks. *IEEE Trans. Neural Netw.* **2011**, *22*, 1588–1598.
18. Ghomi, M.; Goodarzi, M. Peak Load Forecasting of Electric Utilities for West Province of Iran by Using Neural Network without Weather Information. In Proceedings of 12th International Conference on Computer Modelling and Simulation (UKSim), Cambridge, UK, 24–26 March 2010; pp. 28–32.
19. Hernández, L.; Baladrón, C.; Aguiar, J.M.; 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*, 2927–2948.
20. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A.; Lloret, J.; Chinarro, D.; Gómez-Sanz, J.J.; Cook, D. A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants. *IEEE Commun. Mag.* **2013**, *51*, 106–113.