

Chapter 8

Neural Networks Applied to Short Term Load Forecasting: A Case Study

Filipe Rodrigues, Carlos Cardeira and J.M.F. Calado

Abstract A good management of renewable energy systems and energy storage requires Short Term Load Forecasting (STLF). In particular, Artificial Neural Networks (ANN) have proved their ability to cope with data driven nonlinear models. In this paper ANN models are used with input variables such as apartment area, numbers of occupants, electrical appliance consumption and time, in order to achieve a robust model to be used in forecasting energy consumption of general homes. A feed-forward ANN trained with the Levenberg-Marquardt algorithm is tested and their results show a quite accurate model foreseeing that ANNs are a promising tool for STLF.

Keywords Artificial neural networks · Levenberg-Marquardt · Energy forecasting · Hourly and daily energy · Boolean application

F. Rodrigues (✉) · J.M.F. Calado
Departamento de Engenharia Mecânica, IDMEC/ISEL—Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa,
Rua Conselheiro Emídio Navarro, 1959-007 Lisbon, Portugal
e-mail: fmrodrigues@dem.isel.pt

J.M.F. Calado
e-mail: jcalado@dem.isel.ipl.pt

F. Rodrigues
MIT Portugal, Campus IST-Tagus Park, Av. Professor Cavaco Silva,
2744-016 Porto Salvo, Portugal

C. Cardeira · J.M.F. Calado
IDMEC/LAETA, Instituto Superior Técnico, Universidade de Lisboa,
Av. Rovisco Pais, 1049-001 Lisbon, Portugal
e-mail: carlos.cardeira@tecnico.ulisboa.pt

8.1 Introduction

The increased of electricity demand, dioxide carbon emission and the energy resource scarcity are a worldwide concern. This paper seeks to contribute to the sustainable development concept, the rational use and storage of electric power.

The aim of this paper is to show the applicability of an Artificial Neural Network (ANN) approach to develop a simple and reliable method to forecast households' daily and hourly energy consumption for residential and small buildings. ANN favored as a method for forecasting in many applications.

Short-term forecasts, in particular, have become increasingly important since the rise of the competitive energy markets. Most forecasting models and methods have already been tried out on load forecasting, with varying degrees of success.

In contrast with the majority of the works published in this area, this work does not make use of temperature as a network input. Forecasting is made using only past load values and hour of the day. Our objective is to show that if an adequate training set is chosen, it is possible to obtain good results even without considering a huge amount of data like temperature or any other weather data.

The increasing use of renewable energy resources, namely the use of wind power in a larger scale, leads to issues of variable production rates that do not necessary fit to the demand of the consumers. Forecasting the consumers demand becomes of the utmost importance because some of the non-urgent consumer needs of energy (e.g. the turn on or off of the internal cycles of washing machines, air conditioning and fridge equipment) may be relatively shifted (delayed or anticipated) to achieve a better fit of the production profile to the consumption profile without compromising the comfort of the consumer and services level. The use of energy accumulators is always needed but their size can be reduced if a correct forecast of the energy consumption is available.

This work is dedicated to tackle the problem of energy consumption forecasts based in a 18 months long comprehensive set of data obtained from monitoring the real households energy consumption with a 15 mn granularity.

The ANN have been trained and tested using electricity energy consumption data obtained from direct logged in 93 households. A database with consumption records measured between February 2000 and July 2001, in Portugal that was used by Sidler's research team [1], has been used during the current studies. For Short Term Electric Load Forecasting (STELF), as the daily and hourly consumption forecast, this research used inputs that include apartment's areas, inhabitants and electricity energy consumption appliances data, and the daily or the hourly load profile electricity consumption of each household was the output. After the selecting and accept the valid data, all of the daily and hourly energy consumption values of 18 months were used in training and testing the model.

The research analyzed several models, identified and concluded that using MLP neural networks with backpropagation, Levenberg-Marquardt learning algorithm, hyperbolic tangent activation function in input and output layers and 20 neurons in the hidden layer, reveals to be a worthy method to provide acceptable results. For

each network, fraction of variance (R^2), mean absolute percent error (MAPE) and the standard deviation of error (SDE) values were calculated and compared.

This paper is organized as follows: the second section presents the ANN architecture adopted; the third and fourth section explains the load forecasting models, their training processes and provides a deeper discussion about the results achieved. The last section presents the main conclusions and indicates some directions for future work.

8.2 State of the Art

The ability to forecast electricity load requirements is one of the most important aspects of effective management of power systems. The quality of the forecasts directly impacts the economic viability and the reliability of every electricity company. Many important operating decisions such as scheduling of power generation, scheduling of fuel purchasing, maintenance scheduling, and planning for energy transactions are based on electric load forecasting.

There are three different types of electric load forecasting depending on the time horizon and the operating decision that needs to be made: short term, medium term, and long term forecasting. In general, long term forecasting is needed for power system planning, medium term forecasting is needed for maintenance and fuel supply planning, while short term forecasting is needed for the day to day operation of the power system.

The technical literature is abundant with techniques and approaches for performing or improving short term load forecasting. A number of approaches work well with certain power systems or certain geographical areas, while they fail for some other systems due to the nature of the electric load demand: it is complex, highly nonlinear, and dependent on weather, seasonal and social factors.

A number of researchers have compiled extensive surveys on load forecasting. Some of these surveys have been focused on neural networks for short term load forecasting (STLF) [2, 3], some other techniques used such as time series and regression models [4], as well as approaches based on exponentially weighted methods [5], while some other authors provided a general look at all types of load forecasting methodologies [6]. Artificial Neural Networks (ANN) have received a large share of attention and interest. Neural networks are probably the most widely used Artificial Intelligence (AI) technique in load forecasting and papers in the area that have been published since the late 80s. The two main reasons for the use of neural networks as a tool for load forecasting is the fact that neural networks can approach numerically any desired function and the fact of not being dependent upon models. Most studies forecasting with neural networks can be divided into forecasts of an output variable to forecast short-term or multiple outputs to forecast the load curve.

The review of literature mentioned in this paper enabled us to establish in our research a number of variables that are based on the rise in consumption patterns by

households, both during weekdays and weekend, and use the ANN to make forecast the typical values for energy consumption per household for one day, considering a specific type of household.

AlFuhaid et al. [7] use a cascaded neural network for predicting load demands for the next 24 h.

Aydinalp et al. [8] introduced a comprehensive national residential energy consumption model using an ANN methodology. They divided it into three separate models: appliances, lighting and cooling (ALC); Domestic hot water (DHW); and space heating (SH).

Becalli et al. [9] describe an application based on ANN to forecast the daily electric load profiles of a suburban area, using a model based on a multi-layer perceptron (MLP), having as inputs load and weather data. Few years later, Becalli et al. [10] used a forecasting model based on an Elman recurrent neural network to obtain lower prediction error rates assessing the influence of Air Conditioning systems on the electric energy consumption of Palermo, in Italy. The model estimates the electricity consumption for each hour of the day, starting from weather data and electricity demand related to the hour before the hour of the forecast.

Tso and Yau [11] present three modeling techniques for the prediction of electricity energy consumption in two year seasons: summer and winter. The three predictive modeling techniques are multiple regressions, a neural network and decision tree models. When comparing accuracy in predicting electricity energy consumption, it was found that the decision tree model and the neural network approach perform slightly better than the other modeling methodology in the summer and winter seasons, respectively.

Despite the conditional demand analysis (CDA) method [12] is capable of accurately predict the energy consumption in the residential sector as well as others [13–16] at the regional level and national level, however, considering households by households, it was shown that the CDA model has limited utility for modeling the energy consumption in the residential sector.

ANNs are reliable as a forecasting method in many applications, however load forecasting is a difficult task. First, because the load series are complex and exhibits several levels of seasonality: the load at a given hour is dependent not only on the load at the previous hour, but also on the load at the same hour on the previous day, and on the load at the same hour on the day with the same denomination in the previous week. Secondly, there are many important exogenous variables that must be considered, especially weather-related variables. Most authors run their simulations using logged weather values instead of forecasted ones, which is a standard practice in electric load forecasting. However, one should take into account that the forecasting errors in practice will be larger than those obtained in simulations, because of the added weather forecast uncertainty [2].

The work described in this paper was carried out with the aim to achieve forecast approaches for daily and hourly energy consumption of a random household and prediction of several days' energy consumption, using ANNs and a Boolean metering system. The results achieve are encouraging revealing that ANNs are able to forecast daily and hourly energy consumption, as well as a reliable load profile.

To model the electric load profile, it has been used as ANNs training data, hourly and daily data measured at end-use energy consumptions. The energy consumption and electric load profiles have been recorded considering several weeks, including both weekdays and weekend.

8.3 Methodology

8.3.1 Designing Artificial Neural Networks

Selecting an appropriate architecture is in general the first step to take when designing an ANN-based forecasting system. In the current studies, the network architecture was built based on multilayer perceptron (MLP), full-connected, which is a feed-forward type of artificial neural network, and the training task was performed through a backpropagation learning algorithm.

The backpropagation learning algorithms, most common in use in feed-forward ANN, are based on steepest-descent methods that perform stochastic gradient descent on the error surface. Backpropagation is typically applied to multiple layers of neurons and works by calculating the overall error rate of an artificial neural network. The output layer is then analyzed to see the contribution of each of the neurons to that error. The neurons weights and threshold values are then adjusted, according to how much each neuron contributed to the error, to minimize the error in the next iteration. To customize its output, several backpropagation learning algorithms use the Momentum parameter. Momentum can be useful to prevent a training algorithm from getting trapped in a local minimum and determines how much influence the previous learning iteration will have on the current iteration.

In this study, it was used the learning Levenberg-Marquardt algorithm that works as a training algorithm with the capabilities of the pruning methodologies. Pruning is a process of examining a solution network, determining which units are not necessary to the solution and removing those units [17]. By doing the artificial neural network prune the model achieved has reduced complexity and the computational effort to run it, especially in real time, is reduced too.

While backpropagation is a steepest descent algorithm, the Levenberg-Marquardt algorithm is a variation of Newton's method. The advantage of Gauss-Newton over the standard Newton's method is that it does not require calculation of second-order derivatives. The Levenberg-Marquardt algorithm trains an ANN faster (10–100 times) than the usual backpropagation algorithms. The Levenberg-Marquardt algorithm, which consists in finding the update given by:

$$\Delta x = -[J^T(x)J(x) + \mu I]^{-1}J^T(x)e(x) \quad (8.1)$$

where $J(x)$ is the Jacobian matrix, μ is a parameter conveniently modified during the algorithm iterations and $e(x)$ is the error vector. When μ is very small or null the Levenberg-Marquardt algorithm becomes Gauss-Newton, which should provide

faster convergence, however for higher μ values, when the first term within square brackets of Eq. (8.1) is negligible with respect to the second term within square brackets, the algorithm becomes steepest descent. Hence, the Levenberg-Marquardt algorithm provides a nice compromise between the speed of Gauss-Newton and the guaranteed convergence of steepest descent [18].

Therefore, the Levenberg-Marquardt (LM) algorithm combines the robustness of the steepest-descent method with the quadratic convergence rate of the Gauss-Newton method to effectively identify the minimum of a convex objective function. LM outperforms gradient descent and conjugate gradient methods for medium sized nonlinear models. It is a well optimization procedure and it is one that works extremely well in practice. For medium sized networks (a few hundred weights say) this method will be much faster than gradient descent plus momentum and even allows to achieve a model with reduced complexity.

Since one hidden layer is enough to approximate any continuous functions [2], the MLP used in the current studies has three layers: an input layer, a hidden layer and an output layer. The number of neurons used in the input layer depends on the number of input variables being considered as inputs of the forecasting model, while the number of neurons in the output layer depends on the number of variables that one wants to predict with a specific model. Furthermore, the number of neurons in the hidden layer comes down by trial and error procedure to determine a workable number of hidden neurons to use. Selecting a few alternative numbers and then running simulations to find out the one that gave the best fitting (or predictive) performance allows to achieve the number of neurons in the hidden layer.

The activation function for the hidden and output neurons must be differentiable and non-decreasing [2]: as mentioned below in the current studies has been adopted the linear and the hyperbolic tangent functions. The activation functions are used to scale the output given by the neurons in the output layer.

As mentioned above, the ANN architecture was built with three-layer feedforward configuration. In order to achieve the desired performance for the forecasting model obtained, as well as during the artificial neural network training stage, following a trial and error procedure, the network was optimized by varying properties such as learning algorithm, scaling μ , and number of neurons, being the performance evaluated by maximizing the R^2 and minimizing the mean square error (MSE) values. Once defined the final architecture the selected learning algorithm was used during several training cycles to achieve the final connections weights and the bias values. The next step was concerned with test the performance of the model achieved.

McCullouch and Pitts [19] created a computational model for neural networks based on mathematics and algorithms. They called this model threshold logic. The model paved the way for neural network research to split into two distinct approaches. One approach focused on biological processes in the brain and the other focused on the application of neural networks to artificial intelligence.

In the late 1940s psychologist Hebb [20] created a hypothesis of learning based on the mechanism of neural plasticity that is now known as Hebbian learning. Hebbian learning is considered to be a 'typical' unsupervised learning rule and its

later variants were early models for long term potentiation. These ideas started being applied to computational models in 1948 with Turing’s B-type machines.

Farley and Clark [21] first used computational machines, then called calculators, to simulate a Hebbian network at MIT. Other neural network computational machines were created by Rochester et al. [22] and nowadays there are several companies developing software to use ANN, as Mathworks leading developer of mathematical computing software for engineers and scientists, founded in 1984, who has Matlab ANN toolbox.

Figure 8.1 shows a neuron with its external inputs, connections and output.

The MATLAB neural network toolbox (nntool) has been used to train the feedforward ANNs. MATLAB provides built-in transfer functions that have been used for the hidden and output layers as follows: hyperbolic tangent sigmoid (tansig) for the hidden neurons; a pure linear function (purelin) for the output neurons (Fig. 8.2).

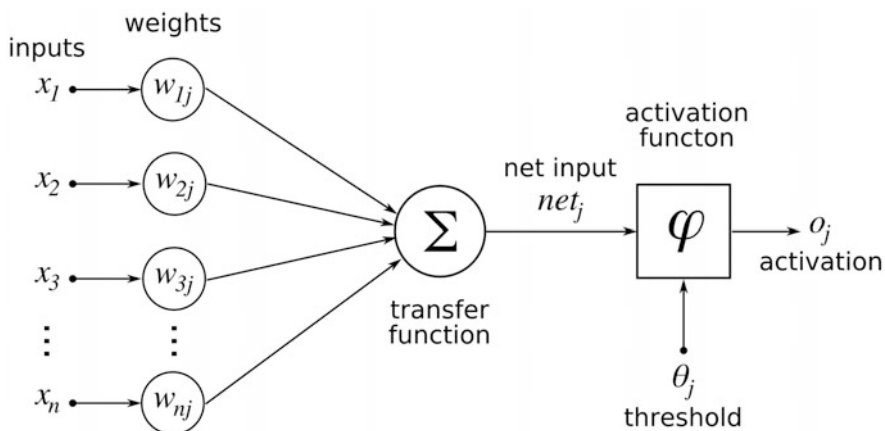


Fig. 8.1 Artificial neuron model

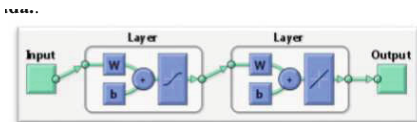


Fig. 8.2 Simplified ANN used with several inputs, hidden neurons with coefficient arrays (W and bias b), and outputs

8.3.2 ANN Energy Consumption Model

Designing ANN based models follows a number of systemic procedures. In general, there are five basic steps: (1) collecting data, (2) pre-processing data, (3) building the network, (4) train, and (5) test the model performance. After data collection, the available data has been pre-processed to train the ANN more efficiently.

ANNs models in this research used the following inputs: electric appliance, as well as apartment area and occupants for a total of 16 inputs. The trained model used the daily electricity consumption recorded data and inputs from a 93 household “training dataset”. The number of people of each household has been selected related with the apartment area as an example of the implication of this method.

This study creates a comprehensive residential energy consumption model using the ANN approach from an energy consumption database consisting of a set of 93 households, recorded in Lisbon (Portugal). Each household have a logged data around 6–8 week, including weekends. Since, ANNs are “data-driven” methods, they typically require large samples during the training stage and they must be appropriately selected. For this study, it was used a complete logged data acquired during six complete weeks at each household for training and validation. A total 93,744 records have been used ($24 \text{ h} \times 7 \text{ days} \times 6 \text{ weeks} \times 93 \text{ households} = 93,744 \text{ h}$).

The mentioned collected data has been pre-processed before being ready to use during the ANN training and testing stages. It has been subjected to some form of preprocessing, that intends to enhance the forecasting model performance, such as “cleaning” the data (by removing outliers, missing values or any irregularities), since during the learning stage the ANN is sensitive to such defective data, degrading the performance of the model obtained.

The missing data are replaced by the preceding or subsequent neighboring values during the same week. Normalization procedure before presenting the input data to the network is generally a good practice, since mixing variables with large magnitudes and small magnitudes will confuse the learning algorithm on the importance of each variable and may force it to finally reject the variable with the smaller magnitude [23].

At the stage of building the network, the researchers specifies the number of hidden layers, neurons in each layer, transfer function in each layer, training function, weight/bias learning function, and performance function. As it is outlined in Sect. 8.4, six different types of training algorithms are investigated for developing the MLP network. During the training network process, the weights are adjusted in order to make the actual outputs (predicated) close to the target (measured) outputs of the network.

This process begun with a network which had 3 neurons in its hidden layer, and repeated, increasing the number of neurons up to 30. The LM algorithm with 20 neurons in the hidden layer for network has produced the best results, and it is used for generating the outputs.

A three-layered feedforward neural network trained by the Levenberg-Marquardt algorithm has been adopted to deliver daily (average and maximum) and hourly

energy forecast (one-step ahead forecasts: forecasts for next day's total load; forecasts for hourly loads and define de load profile from a random day). The inputs to train the ANNs proposed in this paper are historical data load, electric appliance, occupancy and area apartment, which will be detailed below. At the training stage, several numbers of neurons in the hidden layer were tested. The best results were produced with twenty hidden neurons. The output layer has one neuron, which was set up to output load forecasts.

In an ANN the relation between inputs and outputs is driven from the data themselves, through a process of training that consists in the modification of the weights associated to the connections, using learning algorithms. In the learning process a neural network builds an input–output mapping, adjusting the weights and biases at each iteration based on the minimization of some error measure between the output produced and the desired output. Thus, learning entails an optimization process. The error minimization process is repeated until an acceptable criterion for convergence is reached.

Using the backpropagation learning algorithm, the input is passed layer through layer until the final output is calculated, and it is compared to the real output to find the error. The error is then propagated back to the input adjusting the weights and biases in each layer. The standard backpropagation learning algorithm is a steepest descent algorithm that minimizes the sum of square errors.

The standard backpropagation learning algorithm is not efficient numerically and tends to converge slowly. In order to accelerate the learning process, two parameters of the backpropagation algorithm can be adjusted: the learning rate and the momentum. The learning rate is the proportion of error gradient by which the weights should be adjusted. Larger values can give a faster convergence to the minimum but also may produce oscillation around the minimum. The momentum determines the proportion of the change of past weights that should be used in the calculation of the new weights.

To avoid overtraining and overfitting it was used the early stopping method [24] to stop the training process. The generated data set was divided into three subsets: a training set, a validation set, and a test set. A training set is a portion of a data set used to train an ANN for the forecasting. The error on the validation set is monitored during the training process and the last set (test set) of parameters was computed to produce the forecasts.

The validation error will normally decrease during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. The test set error is not used during the training, but it is used as a more objective measure of the performance of various models that have been fitted to the training data.

The evaluation of the ANN performance is based on the shape of the load profile distribution and following criterion: mean absolute percent error (MAPE); the standard deviation of error (SDE) and; serial correlation (linear correlation R and linear regression R^2), are defined as below.

The MAPE criterion is given by:

$$MAPE = \frac{100}{N} \sum_{j=1}^N \frac{|p_j - a_j|}{\bar{p}} \quad (8.2)$$

$$\bar{p} = \frac{1}{N} \sum_{j=1}^N p_j \quad (8.3)$$

Linear correlation coefficient is given by:

$$R = \frac{N \sum_{j=1}^N (p_j a_j) - \sum_{j=1}^N p_j \sum_{j=1}^N a_j}{\sqrt{\left[N \sum_{j=1}^N p_j^2 - \left(\sum_{j=1}^N p_j \right)^2 \right] - \left[N \sum_{j=1}^N a_j^2 - \left(\sum_{j=1}^N a_j \right)^2 \right]}} \quad (8.4)$$

Linear regression is given by:

$$R^2 = 1 - \frac{\sum_{j=1}^N (p_j - a_j)^2}{\sum_{j=1}^N (a_j)^2} \quad (8.5)$$

where p_j and a_j are respectively the forecasted and record load at hour j , \bar{p} is the average price of the forecasting period and N is the number of forecasted hours.

The SDE criterion is given by:

$$SDE = \sqrt{\frac{1}{N} \sum_{j=1}^N (e_j - \bar{e})^2} \quad (8.6)$$

$$e_j = p_j - a_j \quad (8.7)$$

$$\bar{e} = \frac{1}{N} \sum_{j=1}^N e_j \quad (8.8)$$

where e_j is the forecast error at hour j and \bar{e} is the average error of the forecasting period.

Load consumption can rise to tens or even hundreds of times of its normal value at particular hours. It may drop to zero or near at other hours. Hence, the average load was used in Eq. (8.2) to avoid the problem caused by load to zero. In linear correlation (R) and regression (R^2) are statistical measure of how well the correlation or regression line approximates the real data points. An R or R^2 of 1 indicates that the correlation or regression line perfectly fits the data.

The next step is test the performance of the developed model. At this stage unseen data are exposed to the model.

8.3.3 Daily Energy Consumption: Average and Maximum

In order to forecast daily energy consumption average and maximum, the inputs of the network were the daily average energy consumption of each electric appliance (lighting, refrigerator, chest freezer, cooking, dishwasher, washing machine, domestic hot water, cooling and heating systems, TV, VCR/DVD, computers and electronic entertainment) the apartment's areas and the occupants of a training households set (47 households).

To achieve the mentioned model the ANN approach use 16 neurons in the input layer, 20 neurons in the hidden layer and 1 neuron in the output layer (Fig. 8.3). The variations in upper number of hidden neurons did not significantly affect forecasting accuracy [13, 25].

For the daily consumption forecast was used a total of 1581 data, 50% of those data, making up the first subset, were used for ANN training and validation. The last 50% of the original data, making up the second subset, was used to evaluate the prediction capacity of the developed ANN model. The training data from the first subset was used for computing the ANN weights and biases, and the validation, from the second subset, was used to test the accuracy of the ANN model.

The first step for ANN training was take 85% of the data from first subset mentioned above for ANN training and validation. During training, the last 15% was used to evaluate the forecast capacity of the developed ANN model.

Hyperbolic tangent sigmoid function (HTSF) and pure line function (PF) were used as the transfer function in the hidden layer and output layer, respectively, and for the input layer it used PF. All data in input and output layers were normalized in the $[0, 1]$ and weights $[-1, 1]$. In the selected ANN model, inputs were the hourly mean energy consumption of all electric appliance of those direct measurements recorded in those years.

The model was built, trained and tested using several networks each, putting through trial and error a different number of neurons (up to 30) in their hidden layer, using software tool (*nntool*) developed by Matlab. For each trial, the forecasting correlation and error values of the outputs were calculated and compared with the test set (logged data), as well the shape of the load profile distribution.

8.3.4 Hourly Energy Consumption

In order to forecast the hourly energy consumption for a usual day it was used a network with three layers, which is shown in Fig. 8.4. The inputs of the network were the hourly energy consumption of each household, including both weekdays and weekend and the output was the next hourly energy consumption.

An appropriate neural network structure with an adequate initialization method and efficient data processing have been developed to deal and improve nonlinear function and supervised backpropagation algorithm is used to train the networks.

To reduce the complexity of the training task, for the hourly consumption, a total of 3 week data, i.e., 32,760 data ($504 \text{ h} \times 65 \text{ households} = 32,760 \text{ h}$) were used. For each household it was used 504 h, 2/3 of those data (336 h), the two first weeks, making up the first subset, were used for ANN training and validation. The last 1/3 of the original data (168 h), the last week, making up the second subset, was used to evaluate the prediction capacity of the developed ANN model.

The ANN model used Boolean input as an hourly meter system [26] for the forecast hourly load [27]. This method uses an input vector composed of Boolean variable and can take any configuration among 2^n different vectors containing Boolean values such as: 0000...0 = 01:00 h; 1000...0 = 02:00 h; 01000...0 = 03:00 h; 111...1 = 24:00 h. This model uses a concept of 11 data in the input layer: load energy (5 last hour load values and for 6th hour the load 24 h ago) and binary encoding for the time (0 to 24 h) with $n = 5$; 20 neurons in the hidden layer and; 1 output (next hourly load) in the output layer.

The network training was performed by using LM backpropagation algorithms, HTSF were used as the transfer function in the hidden layer and output layer, and PF in input layer, also performed under software tool (nntool) developed by Matlab. All data in input and output layers were normalized in the [0, 1] and weights [-1, 1]. In the selected ANN model, inputs were the real-time hourly energy consumption measurements recorded in those weeks.

For this network, the forecasting correlation and error values of the outputs were calculated and compared with test set, as well the shape of the load profile distribution. The effectiveness of this approach is confirmed by forecast outputs obtained. In conclusion, the proposed artificial neural network shows a reliable performance for load forecasting.

8.4 Validation

8.4.1 Database

Despite the study model and database, electric appliance load profile has been validated by Almeida et al. [28] and based on information gathered from monitoring carried out by Sidler [1], the database was validated by this research team.

The database is characterized by 93,744 hourly energy consumption data collected and recorded directly in 93 households, 18 months, in Lisbon, Portugal.

The researchers analyzed, for each household, the status and profile of each registry and corrected the flaws (w/ value: 0 Wh), by adding value antecedent or precedent (choosing the higher value), thus ensuring the profile of the load curve and mislead the ANN.

Therefore, since the all database has been analyzed in detail and amended, the research team validates the database, since, in addition to the reasons stated in this paper, it ensures consistency of its quality for this work as one comes to demonstrate.

8.4.2 Selection of the Network Architecture

In order to determine the best approach model to forecast energy consumption for each household, it was select several network architecture with one hidden layer and the learning algorithm that produces the best prediction performance. Since there are two input layer (16 and 11 to forecast daily and hourly energy consumption, respectively) and one output layer, for this stage, chosen de neural network architecture, the network sample has 16 inputs, 1 output and 20 neurons, the networks were trained with six learning algorithms available in the Matlab software, available at that time de research forwarded.

Thus, a total of 12 networks (6 learning algorithms × 2 network configurations = 12) were tested. The parameters of the learning algorithms used in the analysis are given in Table 8.1. Training was halted when the testing set sum of squares error (SSE) value stopped decreasing and started to increase—an indication of over-training and de MSE get the best low value—and indication of lower square error. The prediction performance in terms of SSE, R² and MSE of the networks with the lowest testing set SSE and MSE values amongst the 2 networks for each of the six learning algorithms is presented in Table 8.2.

As can be seen from Table 8.2, the feedforward Levenberg-Marquardt produced the best forecast with the higher R² being 0.99962 and faster than the other network with 11 iterations, indicating that this network has the highest performance amongst the networks tested.

By doing this choice, the model structure selection is reduced to dealing with the chosen network inputs/outputs and the internal network architecture. The ANN nonlinear models, expressed by Eq. 8.9, assume that the dynamic process output can be described, at a discrete time instances, by a nonlinear difference equation, depending from the past inputs and outputs [29]:

$$\hat{y}(t) = \hat{f}(y(t - 1), \dots, (t - n), u(t - 1), \dots, u(t - m)) \tag{8.9}$$

where n and m are the output and input signal delays, respectively, and \hat{f} is a nonlinear function (the MLP activation function). Essentially, the MLP networks are used to approximate the \hat{f} function if the inputs of the network (ui(t)) are chosen as the n past outputs and the m past inputs:

Table 8.1 Parameters of the learning algorithms used

| Learning algorithm | Parameters ^a | | | |
|----------------------------------|-------------------------|-----------|-----------------------------|-----------------------|
| Elman backpropagation | μ: 0.001 | μ_dec 0.1 | μ_inc 10 | |
| Layer recurrent | μ: 0.001 | μ_dec 0.1 | μ_inc 10 | |
| NARX | μ: 0.001 | μ_dec 0.1 | μ_inc 10 | |
| Cascade-forward back-propagation | μ: 0.001 | μ_dec 0.1 | μ_inc 10 | |
| Levenberg-Marquardt | μ: 0.001 | μ_dec 0.1 | μ_inc 10 | |
| Resilient propagation | β inc.: 1.2 | β dec 0.5 | φ _{initial} : 0.07 | φ _{max} : 50 |

^athe definitions for the parameters of the learning algorithms are given in the appendix

Table 8.2 The performance of the network trained using six different learning algorithms

| Network | Learning algorithm | No of hidden units | No of neurons | No interaction | SSE | R ² | No interaction | MSE | R ² |
|------------|----------------------------------|--------------------|---------------|----------------|----------------|----------------|----------------|-----------|----------------|
| 1 | Elman back-propagation | 1 | 20 | 14 | 596.000 | - | 91 | 215 | - |
| 2 | Layer recurrent | 1 | 20 | 24 | 403.000 | - | 108 | 143 | - |
| 3 | NARX | 1 | 20 | 27 | 417.000 | - | 29 | 197 | - |
| 4 | Cascade-forward back-propagation | 1 | 20 | 54 | 159.000 | 0.90515 | 12 | 697 | 0.97721 |
| 5 | Feed-forward back-propagation | | | | | | | | |
| 5.1 | Levenberg-Marquardt | 1 | 20 | 167 | 164.000 | 0.99817 | 11 | 28 | 0.99962 |
| 5.2 | Resilient propagation | 1 | 20 | 27 | 66.400.000 | 0.98709 | 89 | 18700 | 0.99796 |

$$\hat{y}_i(t) = F_i \left[\sum_{j=1}^{N_h} v_{ij} f_i \left(\sum_{k=1}^{n+m} w_{jk} x_k(t) + b_{j0} \right) + b_{i0} \right], \quad i = 1, \dots, M \quad (8.10)$$

where N_h is the number of hidden layer neurons [$N_h = 16$ (Fig. 8.3) or 11 (Fig. 8.4)], w_{jk} are the first layer interconnection weights (the first subscript index referring to the neuron, and the second to the input), v_{ij} are the second layer interconnection weights, b_{j0} and b_{i0} are the threshold or sets (biases), and F_i and f_i are the activation functions of the output neurons and hidden neurons, respectively.

Thus, the modeling process is to find the neural network that fits this ANN model equation. To obtain the ANN models some steps must be made:

1. The data collection from the process that are representative of a great part of the process working points;
2. The model structure selection;
3. The model estimation;
4. The model validation.

One of the attractive features of using MLP networks to model unknown non-linear processes is that a separate discretization process is rendered unnecessary since a nonlinear discrete model is available immediately [30].

The MATLAB neural network toolbox was used to train the feedforward backpropagation ANN models, because it had structure selection, estimation, validation and simulation (forecast) in a GUI box. MATLAB provides built-in transfer

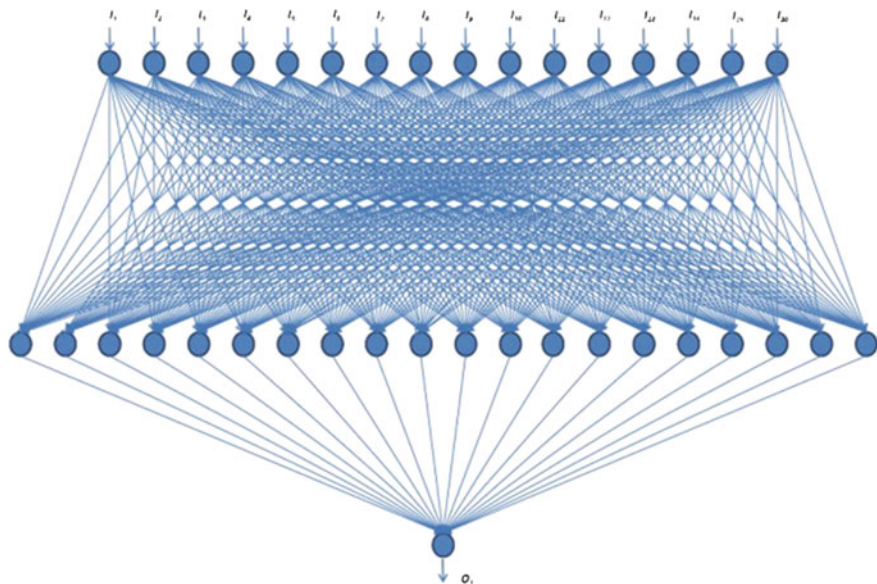


Fig. 8.3 ANN architecture used for daily energy average and maximum consumption

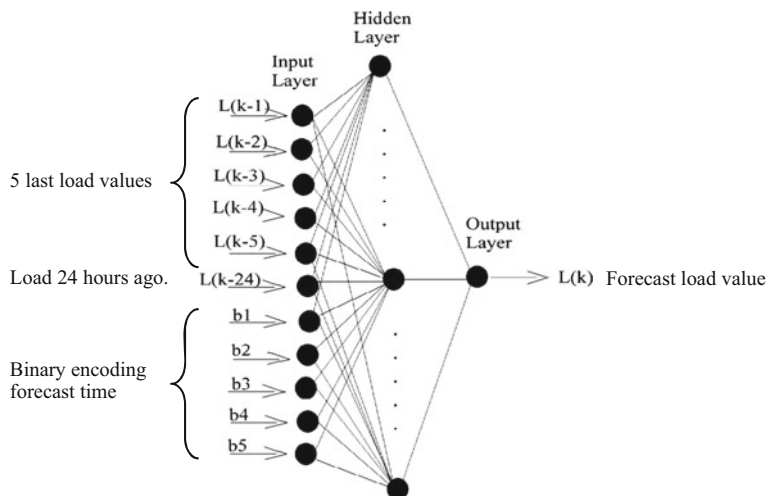


Fig. 8.4 Simplified ANN Architecture used to forecast hourly load

functions which are used in this study: linear (purelin) and Hyperbolic Tangent Sigmoid (HTSF); for the hidden layer and linear functions for the output layer. This function is a nonlinear, continua and differentiable function, as needed for back-propagation training algorithms. The graphical illustration and mathematical form of such functions are shown following:

Linear:

$$\zeta = b + \sum_{k=1}^n w_k x_k \quad (8.11)$$

Hyperbolic Tangent Sigmoid:

$$\zeta = \tanh \left(b + \sum_{k=1}^n w_k x_k \right) \quad (8.12)$$

After this, the determination of the number of hidden neurons has been made by an iterative process. The training process was based on the MSE (Mean square error), the number of neurons in the hidden layer was iteratively reduced to preserve the model performance and, to be the less possible, to ensure generalization capabilities and avoid overfitting.

The algorithm that has been chosen for train the ANN networks models has been the Levenberg-Marquardt algorithm because is the fastest algorithm, and with better results, on function approximation problems as shown in [31]. In general, on function approximation problems, for networks containing up to a few hundred weights, the Levenberg-Marquardt algorithm will have the fastest convergence (but, it needs much more memory).

The neurons activation functions used are the hyperbolic tangent for the hidden layers and linear and hyperbolic tangent function for the output layers, because it is necessary that the ANN weights should be between $[-1, 1]$.

During training the ANN, despite requiring more memory and more time spent for calculation, it was found that hyperbolic tangent activation function provides better performance and lower error (MSE). Thus, it was decided to present the work results with the hyperbolic tangent function.

The feedforward Levenberg-Marquardt algorithm with 20 neurons produced the best forecast with the higher R^2 being 1 and lower MSE with 3.53, indicating that this architecture has the highest performance amongst the architecture tested.

Han et al. [32] used the predicting ANN model 20 neuron that are chosen to match the average number of hidden neuron in the adaptive network.

In futures works, to choose the number of hidden neurons in neural networks, for identification problems, Mendes et al. [31] propose a new approach process of choosing the number of hidden neurons using a pruning algorithm.

8.5 Results and Discussion

The ANN model was trained using the database energy consumption from 93 households, values from years 2000–2001 (18 months), and the trained model was then tested and compared with another set of daily and hourly energy consumption. For learning the neural network, that was adopted the well-known Levenberg–Marquardt back-propagation algorithm and activation function used was the hyperbolic tangent for both hidden and output layers. Also, the training process used the LM algorithm with 20 neurons, because produced the best forecast with the higher R^2 and lower MAPE and SDE. The training of neural network continues unless and until the error becomes constant. After the error becomes constant, the learning procedure terminates.

It started working with the daily average consumption of each household, and concludes with hourly energy consumption. Training models as ANN leads to the conclusion that they were non-linear functions. It found that the proposed artificial neural network (ANN) has an advantage of dealing not only with non-linear part of the load, was able to forecast the daily and hourly energy consumption with good result and precision for a normal day, weekend and special days.

It was tried to find other results with this research, as the neural network in forecasting the maximum consumption for each household, Boolean input application as an hourly meters system and compare the results forecast with real values measured and logged in the database mentioned.

The daily, maximum and hourly consumption has an important role in shaping the energy storage design. Knowing the forecasting consumption, power production and hourly energy demand is possible shedding (anticipate or postpone) the consumption of electricity.

8.5.1 Daily Energy Consumption: Average and Maximum

The neural network model was trained by using data from a random day of each household and tested to forecast a typical day to another household. For each those clusters, to predicted household's daily energy consumption, the results were good. This can be attributed to the amount of daily and hourly training data used to tuning the model [33].

In order to forecast maximum power consumption in a household, it was collected the highest data energy consumption of each day and household. It can be concluded, as the latter work for the daily energy consumption for a random day, the output of ANN with 20 neurons are very close to the real values.

By analyzing the Table 8.3, Figs. 8.5 and 8.6, it is shown that the adopted model for predicting average and maximum consumption of housing allows good outcomes. Correlation values and linear regressions near unity validate the performance and adjustment of results. The values of the error (MAPE) and standard deviation (SDE) confirm sureness in the model adopted. These forecasting are reliable and satisfactory for power demanding and energy storage for each household.

The figures show the modeling results of Portuguese household daily and maximum energy consumption (in the bottom, the horizontal axis, each number identified one household and vertical axis the daily consumption in kWh/day). The blue symbol (*) represents the daily electric energy consumption, measured and recorded in a random day, and the red box (□) line represents the ANN forecast for the same day.

8.5.2 Hourly Energy Demand

Simple Forecast ANN Model

Using the same ANN daily methodology, the following figures show the performance of the neural network to a randomly selected household and it appears that is satisfactory and interesting ANN behaves.

The profiles shown in Figs. 8.7, 8.8, 8.9, 8.10, 8.11, 8.12 and 8.13, allow us to evaluate that are good results early hours forecasting. As one moves away from a week, the performance decreases and the forecast error increased as expected.

Table 8.3 Statistical analysis of the forecasting error attained with the neural network approach for 46 households analyzed

| Daily energy | R (%) | R ² (%) | MAPE (%) | SDE |
|--------------|-------|--------------------|----------|------|
| Average | 98.9 | 97.9 | 4.2 | 1.06 |
| Maximum | 93.5 | 87.4 | 18.1 | 8.25 |

Fig. 8.5 Comparison of daily energy demand [kWh] by household

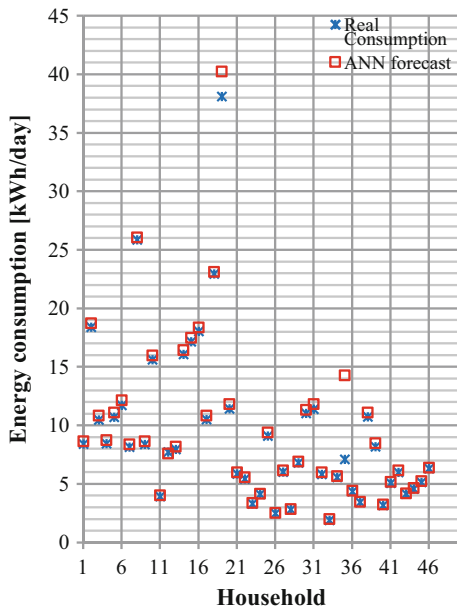
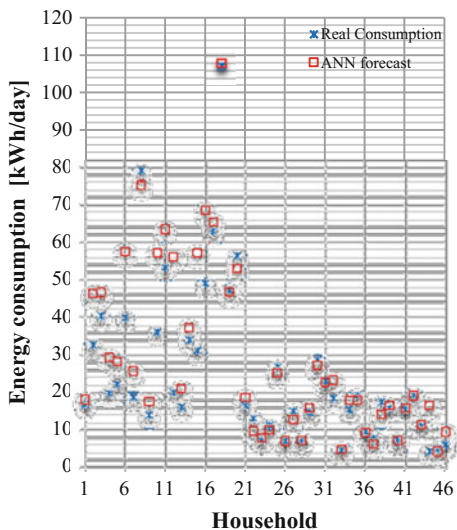


Fig. 8.6 Comparison of maximum consumption [kWh] for each household



Boolean Input Forecast ANN Model

The ANN model used the Boolean input application, with 11 inputs, one hidden and one output layer, as shown in Fig. 8.4, and for learning the neural network, that was adopted the Levenberg–Marquardt algorithm and a network which had 20 neurons in its hidden layer.

Fig. 8.7 ANN—Comparison of hourly consumption—1st day (Saturday)

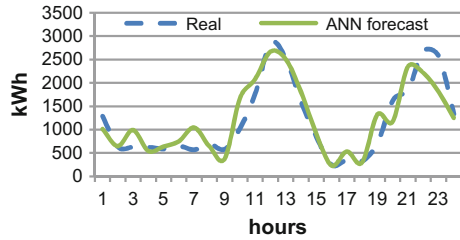


Fig. 8.8 ANN—Comparison of hourly consumption—2nd day (Sunday)

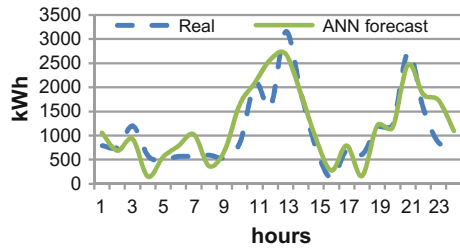


Fig. 8.9 ANN—Comparison of hourly consumption—3rd day (Monday)

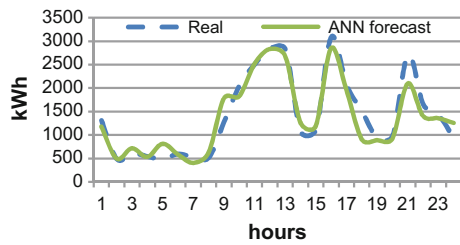
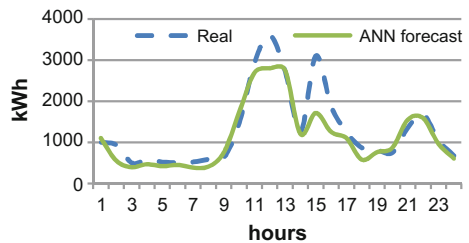


Fig. 8.10 ANN—Comparison of hourly consumption—4th day (Tuesday)



The ANN was trained using the first two weeks of data, one random household and tested for the next 3 day of 3rd week. Figures 8.14 and 8.15 show the forecast hourly load of random households. Evaluating those Figures and Table 8.4 it is found that for daily profiles which have a similar pattern the method proposed by [27, 34] has a good performance for the first three days.

In test process, the forecasting correlation and error values of the outputs were calculated are:

Fig. 8.11 ANN—
Comparison of hourly
consumption—5th day
(Wednesday)

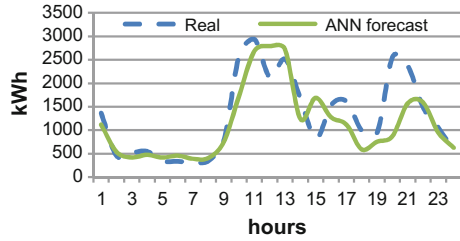


Fig. 8.12 ANN—
Comparison of hourly
consumption—6th day
(Thursday)

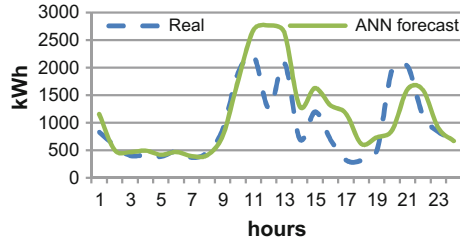


Fig. 8.13 ANN—
Comparison of hourly
consumption—7th day
(Friday)

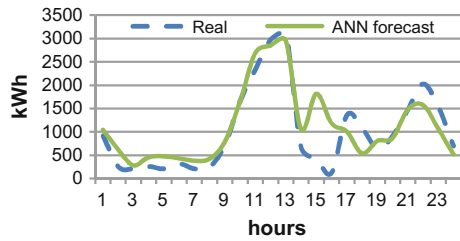


Fig. 8.14 Forecast hourly
load of household no 64—
first 3 days of 3rd week

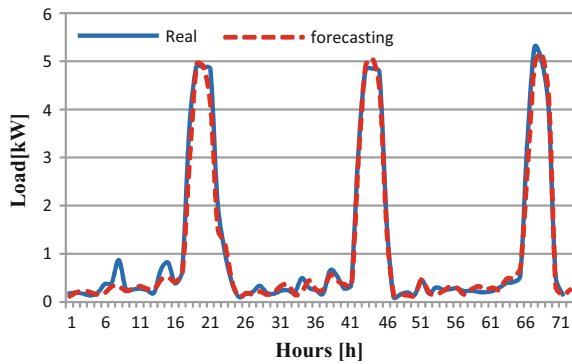


Fig. 8.15 Forecast hourly load of household no 65—first 3 days of 3rd week

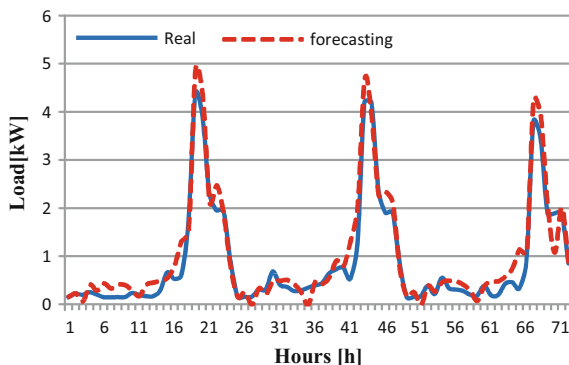


Table 8.4 Statistical analysis of the forecasting error attained with the neural network approach for the households H64 (left) and H65 (right) analysed

| | R (%) | R ² (%) | MAPE (%) | SDE | R (%) | R ² (%) | MAPE (%) | SDE |
|-----|-------|--------------------|----------|------|-------|--------------------|----------|------|
| 1st | 98.8 | 97.7 | 16.0 | 0.23 | 98.3% | 96.6% | 21.5 | 0.17 |
| 2nd | 99.6 | 99.3 | 10.0 | 0.09 | 97.6% | 94.5% | 19.3 | 0.21 |
| 3rd | 99.3 | 98.6 | 12.9 | 0.14 | 97.9% | 93.2% | 23.5 | 0.22 |

8.5.3 Results Obtained in Similar Studies

The analysis of the current literature on the use of MLP neural networks with backpropagation forecasting a daily load curve with an advance of 24 h, reveals a small number of papers mostly dedicated to large power systems operation as electric companies. These methods are able to provide good results and the use of temperature ensures an improved quality of forecasts [35–38].

However, comparing the work shown in this paper with the existing works in the scientific literature, the approach followed in this work forecasts electric load for long periods ahead (i.e., over 24 h), temperature and other weather data are not considered in these predictions and it was not used the group of days (weekday and weekend). The neural network used in this work is fed with actual load values available in order to forecast the next period. In [34] a 30 days forecast (following month) using data of 30 days (preceding month) is shown. The present approach shows a 3 days forecast using 15 days (two weeks) data with an ANN that is simpler providing results that present the same level of accuracy.

The figures show the modeling results of Portuguese household hourly energy consumption (in the bottom, the horizontal axis, each number identified one hour and vertical axis the energy consumption in kWh). The straight blue line represents the hourly load, measured and recorded in the first 3 days of the 3rd week, and the discontinuous red line represents the ANN forecast for the same days.

8.6 Conclusion

As current literature, these results show that the ANN technique can be reliably used in forecasting daily and hourly electric energy consumption and evaluated the performance of the neural network for forecasting the hourly energy demand. It will adjust the network and make the forecast of precedent hourly consumption for each household.

This paper introduced forecasting methods of daily and hourly energy consumption, by using an Artificial Neural Network (ANN). After identifying a faster algorithm, such as Levenberg-Marquardt, the research reveal that ANN are able to forecast daily and hourly energy consumption, as well the load profile with accuracy. This can be attributed to the amount of daily and hourly training data used to tuning the model [33].

Cluster analysis method has been applied based on electric consumption and occupancy patterns. To determinate the electric load profile, the required input data are hourly and daily data measured end-use energy consumptions. The daily energy consumption and electric load profiles have been determined using several weeks, including both weekdays and weekend.

The paper introduced the ANN technique for modeling energy consumption for a random day, the next hours (until 3rd day), using a Boolean metering system. To determinate the electric load profile, the required input data are hourly and daily data logged end-use energy consumptions. The energy consumption and electric load profiles have been determined using several weeks, including both weekdays and weekend.

The apartment area and inhabitants inputs are the two variables it define the shape of the forecasting. The input data as kitchen appliances, lighting, cooling and heating, domestic hot water (DHW) and entertainment appliances have negligible contribution to the improvement of the forecasting performances. This analysis allows concluding that, in the region of Lisbon, the households are provided with similar electrical appliances. However, this extrapolation to other regions and countries should be carefully evaluated.

Forecast hourly and daily energy consumption can be useful to determine the required size of a storage energy system, delay and postpone energy consumption, and can be used at the renewable energy system early design stage. It can also help on the demand-side management (DSM), such as electricity suppliers, to forecast the likely future development of electricity demand in the entire sector of the community.

For future research, an important step is continuing this additional research enhancing forecasting capability on the load profile renewable energy production (micro production) to identify an accurate, effective and appropriate renewable energy production and energy storage.

Acknowledgements This work was partially supported by FCT, through IDMEC, under LAETA PestOE/EME/LA0022 and by the Sustainable Urban Energy System (SUES) project under the MIT Portugal Program (MPP).

References

1. Sidler, O.: Demand-side management end-use metering campaign in the residential sector, European community SAVE programme (2002)
2. Hippert, H.S., Pedreira, C.E., Souza, C.R.: Neural networks for short-term load forecasting: a review and evaluation. *IEEE Trans. Power Syst.* **16**(1), 44–55 (2001)
3. Carpinteiro, O., Reis, A., Silva, A.: A hierarchical neural model in short-term load forecasting. *Appl. Soft Comput.* **4**, 405–412 (2004)
4. Gross, G., Galiana, F.D.: Short-term load forecasting. *Proc. IEEE* **75**(12), 1558–1573 (1987)
5. Taylor, J.W.: Short-term load forecasting with exponentially weighted methods. *IEEE Trans. Power Syst.* **27**, 458–646 (2012)
6. Feinberg, E.A., Genethliou, D.: Load forecasting. *Applied Mathematics for Restructured Electric Power Systems: Optimization Control, and Computational Intelligence*, vol. Chapter 12 (2005)
7. AlFuhaid, A.S., El-Sayed, M.A., Mahmoud, M.S.: Cascaded artificial neural networks for short-term load forecasting. *IEEE Trans. Power Syst.* **12**(4), 1524–1529 (1997)
8. Aydinalp, M., Ugursal, V., Fung, A.: Modeling of the appliance, lighting, and space cooling energy consumptions in the residential sector using neural networks. *Appl. Energy* **72**(2), 87–110 (2002)
9. Beccali, M., Cellura, M., Lo Brano, V., Marvuglia, A.: Forecasting daily urban electric load profiles using artificial neural networks. *Energy Convers. Manage.* **45**, 2879–2900 (2004)
10. Beccali, M., Cellura, M., Lo Brano, V., Marvuglia, A.: Short-term prediction of household electricity consumption: Assessing weather sensitivity in a Mediterranean area. *Renew. Sustain. Energy Rev.* **12**, 2040–2065 (2008)
11. Tso, G., Yau, K.: Predicting electricity energy consumption: a comparison of regression analysis, decision tree and neural networks. *Energy* **28**, 1761–1768 (2005)
12. Aydinalp, M., Ugursal, V.: Comparison of neural network, conditional demand analysis, and engineering approaches for modeling end-use energy consumption in the residential sector. *Appl. Energy* **85**, 271–296 (2008)
13. Khotanzad, A., Afkhami-Rohani, R., Abaye, A., Davis, M., Maratukulam, D.: ANNSTLF—a neural-network-based electric load forecasting system. *IEEE Trans. Neural Networks* **8**(4), 835–846 (1997)
14. Kalogirou, S.A.: Applications of artificial neural networks in energy systems: a review. *Energy Convers. Manage.* **1999**(40), 1073–1087 (1999)
15. Kalogirou, S.A.: Applications of artificial neural-networks for energy systems. *Appl. Energy* **67**(1–2), 17–35 (2000)
16. Kalogirou, S.A.: Artificial neural networks energy systems applications. *Renew. Sustain. Energy Rev.* **5**, 373–401 (2001)
17. Sietsma, J., Dow, R.: Neural net pruning: why and how. In: *Proceedings of IEEE International Conference on Neural Networks*, vol. 1, pp. 325–333 (1988)
18. Saini, L.M., Soni, M.K.: Artificial neural network based peak load forecasting using Levenberg-Marquardt and quasi-Newton methods. *IEE Proc. Gen. Transm. Distrib.* **149**(5), 578–584 (2002)
19. McCulloch, W., Pitts, W.: A logical calculus of ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**(4), 115–133 (1943)
20. Hebb, D.: *The Organization of Behavior* (1949)
21. Farley, B., Clark, W.A.: Simulation of self-organizing systems by digital computer. *IRE Trans. Inf. Theor.* **4**(4), 76–84 (1954)
22. Rochester, N., Holland, J.H., Habit, L.H., Duda, W.L.: Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Trans. Inf. Theor.* **2**(3), 80–93 (1956)
23. Tymvios, F., Michaelides, S., Skouteli, C.: Estimation of surface solar radiation with artificial neural networks. In: *Badescu, V. (ed.) Modeling Solar Radiation at the Earth's Surface: Recent Advances*, p. 221–256 (2008)

24. Sarle, W.S.: Stopped training and other remedies for overfitting. In: Proceedings of the 27th Symposium on the Interface of Computing Science and Statistic, pp. 362–360 (1995)
25. Bakirtzis, A.G., Petridis, V., Kiartzis, S.J., Alexiadis, M.C.: A neural network short term load forecasting model for the Greek power system. *IEEE Trans. Power Syst.* **11**(2), 858–863 (1996)
26. W. Holderbaum, Canart, R., Borne, P.: Artificial neural networks application to boolean input systems control. Laboratoire d'Automatique et d'Informatique Industrielle de Lille (1999)
27. Lacerda, W.S.: Conference at Universidade Federal de Lavras (UFLA), Nov 2006
28. Almeida, A., Fonseca, P., Schlomann, B., Feilberg, N., Ferreira, C.: Residential monitoring to decrease energy use and carbon emissions in Europe. Presented in international energy efficiency in domestic appliances and lighting conference 2006, 2006
29. Mendes, M.J.G.C., Calado, J.M.F., Sá da Costa, J.M.G.: Multi-agent approach to fault tolerant control systems. Tese de Doutoramento em Engenharia Mecânica (em inglês) (2008)
30. Norgaard, M., Ravn, M.O., Poulsen, N.K., Hansen, L.K.: *Neural Networks for Modelling and Control of Dynamic Systems*. Springer (2000)
31. Mendes, J.G.C., Calado, J.M.F., Sá da Costa, J.M.G.: Pruning algorithm applied to a hierarchical structure of fuzzy neural networks: case study. In: 8th IEEE International Conference on Methods and Models in Automation and Robotics—MMAR 2002, vol. 1, p. 207–212. 2–5 Sept 2002
32. Han, H.G., Wu, X.L., Qiao, J.F.: Real-time model predictive control using a self-organizing neural network. *IEEE Trans. Neural Networks Learn. Syst.* **24**(9) 1425–1436 (2013)
33. Mikhalakakou, G., Santamouris, M., Tsangrassoulis, A.: On the energy consumption in residential buildings. *Energy Buildings* **34**(7), 727–736 (2001)
34. Zebulum, R.S., Vellasco, M., Guedes, K., Pacheco, M.A.: Short-term load forecasting using neural nets. In: Proceedings Natural to Artificial Neural Computation, Springer, Berlin—International Workshop on Artificial Neural Networks Malaga-Torremolinos, vol. 930, ISBN 978-3-540-49288-7, pp. 1001–1008. 7–9 June 1995
35. Afkhami, R., Yazdi, F.M.: Application of neural networks for short-term load forecasting. In: IEEE Power India Conference, 10–12 Apr 2006
36. He, Y.-J., Zhu, Y.-C., Gu, J.-C., Yin, C.-Q.: Similar day selecting based neural network model and its application in short-term load forecasting. In: Proceedings of International Conference on Machine Learning and Cybernetics, vol. 8, pp. 4760–4763. 18–21 Aug 2005
37. Ramezani, M., Falaghi, H., Haghifam, M.R., Shahryari, G.A.: Short-term electric load forecasting using neural networks. In: The International Conference on Computer as a Tool—EUROCON 2005, vol. 2, pp. 1525–1528. 21–24 Nov 2005
38. Barghinia, S., Ansarimehr, P., Habibi, H., Vafadar, N.: Short term load forecasting of Iran national power system using artificial neural network. In: Power Tech Proceedings, 2001 IEEE Porto, vol. 3, 10–13 Sept 2001