Energy Conversion and Management 123 (2016) 445–452

Contents lists available at ScienceDirect

Energy Conversion and Management

journal homepage: www.elsevier.com/locate/enconman

Robust total energy demand estimation with a hybrid Variable Neighborhood Search – Extreme Learning Machine algorithm

J. Sánchez-Oro [a], A. Duarte [a], S. Salcedo-Sanz [b,*]

[a] Dept. of Computer Science, Universidad Rey Juan Carlos, Móstoles, Spain
[b] Dept. of Signal Processing and Communications, Universidad de Alcalá, Madrid, Spain

ARTICLE INFO

Article history:
Received 1 February 2016
Received in revised form 21 May 2016
Accepted 17 June 2016
Available online 29 June 2016

Keywords:
Energy demand estimation
Variable Neighborhood Search
Extreme Learning Machines
Socio-economic predictive variables

ABSTRACT

Energy demand prediction is an important problem whose solution is evaluated by policy makers in order to take key decisions affecting the economy of a country. A number of previous approaches to improve the quality of this estimation have been proposed in the last decade, the majority of them applying different machine learning techniques. In this paper, the performance of a robust hybrid approach, composed of a Variable Neighborhood Search algorithm and a new class of neural network called Extreme Learning Machine, is discussed. The Variable Neighborhood Search algorithm is focused on obtaining the most relevant features among the set of initial ones, by including an exponential prediction model. While previous approaches consider that the number of macroeconomic variables used for prediction is a parameter of the algorithm (i.e., it is fixed a priori), the proposed Variable Neighborhood Search method optimizes both: the number of variables and the best ones. After this first step of feature selection, an Extreme Learning Machine network is applied to obtain the final energy demand prediction. Experiments in a real case of energy demand estimation in Spain show the excellent performance of the proposed approach. In particular, the whole method obtains an estimation of the energy demand with an error lower than 2%, even when considering the crisis years, which are a real challenge.

1. Introduction

Total energy demand forecasting in a country is an important problem faced by decision makers every year [1]. In fact, energy demand has increased sharply at a worldwide scale, mainly due to an aggressive industrialization of developed and developing countries in the last 3 decades. Just in this respect, industrialization consumes over 50% of the total energy demand in developing countries, alternatively, the rapid population growth and globalization also contribute to the high energy consumption all around the world.

In this context it is well known that, as the economy grows, the energy demand increases exponentially, what brings along important environmental issues that may compromise the future of next generations. Currently, 80% of the energy demand in the world is covered by non-renewable energy sources such as coal or petroleum, with more dramatic values of this indicator foreseen at developing countries. Consequently, countries with a growing industrial activity happen to be more energy demanding than others with economies based on alternative sectors. In this context, predicting medium and long-term energy demand is a key problem faced by policy makers, with impact in all countries' economies and nations' development [2].

Different previous approaches have tackled this important problem of energy demand estimation, from different perspectives. Interestingly, many of the previous approaches have been focused on estimating energy demand in emerging economies such as Turkey. The most important ones are described here: one of the first works dealing with a problem of energy demand estimation in Turkey is [3], where a genetic algorithm to optimize the parameters of an exponential prediction model has been proposed. Four socio-economic variables such as Gross Domestic Product, population and imports and exports figures were used to carry out this energy demand prediction, with annual resolution, and future projection for all the predictive variables and the energy demand in the country. From that initial work, there have been very different approaches focused on small variations of this problem, also in case studies in Turkey. In [4] a problem of primary energy demand is tackled, with a classical ARIMA forecasting approach (box-jenkins methodology). The authors presented as well future

* Corresponding author.
E-mail addresses: jesus.sanchezoro@urjc.es (J. Sánchez-Oro), abraham.duarte@urjc.es (A. Duarte), sancho.salcedo@uah.es (S. Salcedo-Sanz).

http://dx.doi.org/10.1016/j.enconman.2016.06.050
0196-8904/

Nomenclature

Acronyms

ACO	Ant Colony Optimization
BVNS	Basic Variable Neighborhood Search
ELM	Extreme Learning Machine
FS	Feature Selection
GA	Genetic Algorithm
GDP	Gross Domestic Product
GVNS	Variable Neighborhood Search
HS	Harmony Search
kTOE	kilotonne of oil equivalent
LB	Lower Bound
MAE	Mean Average Error
MLP	Multi-Layer Perceptron
PSO	Particle Swarm Optimization
RVNS	Reduced Variable Neighborhood Search
SVNS	Skewed Variable Neighborhood Search
UB	Upper Bound
VND	Variable Neighborhood Descent
VNDS	Variable Neighborhood Decomposition Search
VNS	Variable Neighborhood Change

scenarios for energy demand based on projections of the predictive variables. The deep global crisis of 2008 made these projections completely fail to reproduce the real behavior of energy demand. In [5] an Ant Colony Optimization (ACO) approach was presented in a similar problem of energy demand estimation in Turkey. In this case the system had to optimize the parameters of a quadratic estimation model for the energy demand prediction, and the predictive variables were the same as in [3]. A similar approach with ACO algorithm has been presented in [6] for the estimation of electricity energy generation and demand in Turkey, also with a quadratic prediction model and socio-economic predictive variables. In [7] a Particle Swarm approach (PSO) is proposed for this problem, including two prediction models, a linear and a quadratic one. Five predictive variables are considered in that work, the ones in [3] plus the country's growth rate in the years of analysis. In [8] a PSO algorithm is also applied, in this case to a problem of electricity demand in Turkey, considering linear and quadratic models. Soon later, the same authors proposed a prediction model based on a hybrid PSO and ACO algorithms in [9], with similar predictive variables and models. In [10] three different models for energy consumption have been proposed: multiple-linear regression analysis, power regression analysis and artificial neural networks. The predictive variables considered are socio-economic variables similar to the ones used in previous approaches, plus unemployment rate. In [11] artificial neural networks are applied to study energy dependence of Turkey. In this case a larger number of socio-economic predictive variables are considered, and the effect of the crisis is also taken into account to make future projections of energy demand in the country.

Alternative approaches have been also applied to energy demand estimation in other countries. In [12] an approach based on neural networks (multi-layer perceptrons) have been considered for South Korea, using same predictive variables that in [3]. A similar approach based on neural networks for the case of energy demand estimation in Greece has been presented in [13]. A comparison with kernel methods has been carried out in that paper, showing good results on the neural networks' performance. Recently, some works have been published on the application of a hybrid PSO-GA algorithm in energy demand estimation problems in China. Specifically, in [14] a PSO-GA has been applied to a problem of energy demand forecasting in China. Three different prediction models, linear, exponential and quadratic are considered, and the predictive variables are economic growth, population, economic structure, urbanization rate, energy structure and energy price. Two similar approaches with the same algorithm and different objective variable has been subsequently published: [15], where a problem of energy demand future projection has been considered, and [16] where energy demand forecasting in the primary section has been considered. In [17] several new models based on logarithmic and alternative exponential functions are used, optimized by a real-encoding evolutionary algorithm for energy demand estimation for metal industry in Iran. More recently, in [18], a Harmony Search (HS) approach with feature selection has been proposed for a problem of energy demand estimation in Spain. In that paper the HS algorithm looks for the best set of features and the optimal weights for an exponential model, in order to obtain a model for energy demand estimation in Spain. An initial pool of 14 predictive variables is considered, and the objective is to estimate the energy demand with an annual time horizon prediction. A second prediction problem with CO_2 as objective variable is also explored in that paper. In [19] a problem of electricity demand forecasting in Italy is considered. Linear regression models have been applied in this case. A similar approach for the case of New Zealand, with multiple linear regression algorithms, with socio-economic and demographic variables has been tackled in [20].

This paper discusses a problem of one-year-ahead energy demand estimation in Spain from socio-economic variables, using a novel hybrid algorithmic approach for optimization and prediction. First a Variable Neighborhood Search (VNS) approach is considered to find the most relevant features among the set of available ones, with an exponential prediction model similar to the one proposed in [3]. On the other hand, an Extreme Learning Machine (ELM) neural network makes use of the feature selection performed by the VNS to carry out the final energy demand prediction. The paper details both algorithms, and the different adaptations that are implemented, mainly in the VNS, to improve the prediction capability of the proposed hybrid approach. Experiments in a real case of energy demand prediction from socio-economic predictive variables in Spain are carried out. It is worth mentioning that previous approaches consider that the number of features is set to a fixed value. In other words, if this number is p, the corresponding optimization algorithm selects the best p features among the whole set of features. However, p should be also a parameter to be optimized since there is not *a priori* information to determine it. The main contribution of this paper is not assuming this starting hypothesis, considering p as parameter to be also optimized. In particular, the proposed VNS finds an undetermined number of features that minimizes the difference between prediction and the actual energy demand. A new methodological contribution to the VNS framework is also proposed in this work: given a solution to an optimization problem, traditional VNS algorithms navigate over the search space by means of perturbations (shake procedure) and improvements (local search). These two operations are applied over the whole solution. In this paper, a new mechanism where the perturbation only affects to a part of the solution (feature selection), while the improvement stage is focused on the other part of the solution (parameter adjustment), is explored for the prediction problem at hand.

The remainder of the paper is structured as follows: Section 2 describes the problem of energy demand estimation tackled in this work. Section 3 presents the proposed approach, by describing the main characteristics of the VNS and ELM algorithms. Section 4 describes the experimental evaluation of the proposed algorithms, where the performance of the proposed approach is compared to that of alternative algorithms for prediction of energy demand. Finally, Section 5 closes the paper with some ending conclusions and remarks.

2. Problem definition

Let us consider a time series $\mathbf{E} \triangleq \{E(t)\}_{t=1}^{n}$ of past energy demands for a given country, with n discrete values corresponding to different years; and a set of m predictive variables $\mathbf{X} = \{X_1(t), \ldots, X_m(t)\}$, with $t = 1, \ldots, n$. A model $\mathcal{M}$ provides an estimation $\hat{\mathbf{E}}$ for $\mathbf{E}$. The problem tackled in this paper consists of finding both: the best subset $\mathbf{X}' \subseteq \mathbf{X}$ of $m' \leqslant m$ features out of the m possible variables in $\mathbf{X}$; and the values $\mathbf{W}$ for the parameters of the model $\mathcal{M}$, where $\mathbf{W} = \{\epsilon, \alpha_1, \ldots, \alpha_{m'}, \beta_1, \ldots, \beta_{m'}\}, \epsilon$ is a bias, which is not related to any feature selected, and the pair α_i, β_i represents the coefficients of feature X'_i with $1 \leqslant i \leqslant m'$. For this problem, the model $\mathcal{M}$ used to estimate the energy demand is an exponential model as was previously suggested in [3,18]. More formally,

$$\hat{E}(t+1) = \epsilon + \sum_{i=1}^{m'} \alpha_i X'_i(t)^{\beta_i} \tag{1}$$

The quality of a solution $S = (\mathbf{X}', \mathbf{W})$ is evaluated using a given objective function, usually related to the similarity of the model output to the real energy demand values. In this case, it is considered that the mean squared error between the observed values and the predicted ones, which is to be minimized. In mathematical terms:

$$f(S) = \frac{1}{n^\star} \sum_{j=1}^{n^\star} \left(E(j) - \hat{E}(j)\right)^2, \tag{2}$$

where $n^\star$ is the size of a reduced training sample ($n^\star < n$).

For the sake of clarity, a solution S of this model is represented by using two different parts, **X'** and **W**. The first one refers to the subset of selected features in $\mathcal{M}$ (as it was previously introduced). For instance, $\mathbf{X}' = \{X_1, X_3, X_4, X_5\}$ represents a solution where features $X'_1 = X_1, X'_2 = X_3, X'_3 = X_4$, and $X'_4 = X_5$, are selected from the whole set $\mathbf{X}$ of m features. Notice that the remaining $m - |\mathbf{X}'|$ features are discarded in this particular solution. Once the subset of selected features in $\mathcal{M}$ have been selected, we need to provide the values for the coefficients of the model.

An example will illustrate how to evaluate a solution by using the aforementioned representation. Let us consider the solution $S = (\mathbf{X}', \mathbf{W})$, where $\mathbf{X}' = \{X_1, X_3, X_4, X_5\}$, and $\mathbf{W} = \{3.4, 0.1, -0.05, 0.3, 0.01, -0.2, -0.1, 0.05, 0.33\}$. As it can seen, for each feature X'_i, a pair of coefficients, α_i and β_i, plus the first bias coefficient, ϵ, are provided. If the values of the selected features for a given year are $X_1 = -0.2, X_3 = 0.1, X_4 = 0.7, X_5 = -0.4$, then they can be used in the model to predict the energy demand for the next year ($t + 1$) as follows:

$$\begin{aligned}
\hat{E}(t+1) &= \epsilon + \alpha_1 \cdot X'_1(t)^{\beta_1} + \alpha_2 \cdot X'_2(t)^{\beta_2} + \alpha_3 \cdot X'_3(t)^{\beta_3} + \alpha_4 \cdot X'_4(t)^{\beta_4} \\
&= \epsilon + \alpha_1 \cdot X_1(t)^{\beta_1} + \alpha_2 \cdot X_3(t)^{\beta_2} + \alpha_3 \cdot X_4(t)^{\beta_3} + \alpha_4 \cdot X_5(t)^{\beta_4} \\
&= 3.4 + 0.1 \cdot (-0.2)^{-0.05} + 0.3 \cdot 0.1^{0.01} - 0.2 \cdot 0.7^{-0.1} \\
&\quad + 0.05 \cdot (-0.4)^{0.33} \\
&= 3.34
\end{aligned}$$

It is important to remark that each feature is normalized in the range $[-1, 1]$ (as it is customary in this context). Additionally, and to avoid scale problems, each parameter of the model takes on a value in the interval $[-1, 1]$. Finally, the bias parameter ϵ considers a larger range of values, $[-5, 5]$, in order to increase the margin of the model to obtain a better fit.

The above formulation corresponds to a class of the so-called Feature Selection (FS) problem. It is an important task in supervised classification and regression problems because irrelevant features, used as part of the training procedure can unnecessarily increase the cost and running time of a prediction system, as well as degrade its generalization performance [21].

3. Materials and methods

This paper proposes the use of two different artificial intelligence approaches in order to perform the one-year-ahead energy demand prediction. The first one is a meta-heuristic named Variable Neighborhood Search (VNS), which consists of a systematic exploration of different predefined neighborhoods. The second one is a Extreme Learning Machine (ELM), a neural network characterized by a fast training stage. Each of the proposed methods focuses on different goals. On one hand, VNS is used to find the most relevant features among the set of available ones. On the other hand, ELM makes use of the feature selection performed by VNS to perform the energy demand prediction. In the next subsections, detailed description of the VSN and ELM approaches are given, and how they have been applied to this specific problem of energy demand prediction.

3.1. Variable Neighborhood Search

Variable Neighborhood Search (VNS) is a metaheuristic originally proposed [22] as a general framework for solving hard optimization problems. VNS is based on performing systematic changes of neighborhood during the search space exploration, in order to escape from local optima. This methodology is in constant evolution, which has resulted in a large variety of strategies, where the most relevant are Reduced VNS (RVNS), Variable Neighborhood Descent (VND), Basic VNS (BVNS), Skewed VNS (SVNS), General VNS (GVNS) or Variable Neighborhood Decomposition Search (VNDS). A complete survey on VNS has been recently presented in [23]. Some recent successful applications of VNS to solve hard optimization problems are [24], where the Cutwidth Minimization Problem is tackled, [25], where the VNS is applied to compute optimal graph separators or [26], which describes an application of the VNS to the Vertex Separation Problem.

In the problem tackled in this work, the quality of a selected subset of features, $\mathbf{X}'$, cannot be evaluated without adjusting the parameters of the model, $\mathbf{W}$. Therefore, there is not *a priori* information available about which features are suitable to be included in the solution. Furthermore, the features are not independent, since the result of including a feature in a solution depends on which other features have been already included on it. Thus it is not easy to design a local search method that operates over the whole solution, but only on part of it. This paper focuses on the

Basic VNS (BVNS) variant, but applying the local search method only to a part of the solution. Algorithm 1 presents the pseudo-code of the proposed algorithm.

Algorithm 1. Basic Variable Neighborhood Search $(S, k_{\max})$

1: **while not** *StoppingCriterion* **do**
2: $k = 1$
3: **while** $k \leqslant k_{\max}$ **do**
4: $S_s \leftarrow Shake(S, k)$
5: $S_{ls} \leftarrow LocalSearch(S_s)$
6: **if** $f(S_{ls}) < f(S_s)$ **then**
7: $S \leftarrow S_{ls}$
8: $k \leftarrow 1$
9: **else**
10: $k \leftarrow k + 1$
11: **end if**
12: **end while**
13: **end while**

The method needs two different parameters: S and $k_{\max}$. The former represents the initial solution, while the latter refers to the maximum neighborhood to be explored. The initial solution considered is constructed by randomly selecting a subset of features. Starting from the first neighborhood (step 2), the algorithm iterates until reaching the largest neighborhood allowed (steps 3–12). In each iteration, BVNS randomly perturbs the incumbent solution to obtain a new solution S_s in the current neighborhood. After that, the local search procedure is applied to the perturbed solution, generating an improved solution S_{ls}. Finally, if S_{ls} outperforms the incumbent one, the search starts again from the first neighborhood ($k = 1$, updating the best solution found). Otherwise, the search continues with the next neighborhood ($k = k + 1$) until reaching the largest neighborhood $k_{\max}$. The algorithm stops when reaching the stopping criterion (steps 1–13). There are different criteria available to stop a stochastic optimization algorithm [27]. Tolerance, number of function evaluations, and maximum number of iterations are some examples. In this case, the number of iterations has been selected as stopping criterion. It is important to remark that the perturbation is performed in the selected features, $\mathbf{X}'$, while the local search method is applied over the parameters of the model, $\mathbf{W}$.

The perturbation performed is described in Algorithm 2. Given the perturbation size (k), the method modifies k features at random (steps 1–8). Then, for each randomly selected feature (step 2), the method either adds it to the solution if it is not included (step 4), or removes it otherwise (step 6).

Algorithm 2. Shake$(S = (\mathbf{X}', \mathbf{W}), k)$

1: **for** $i \in 1 \ldots k$ **do**
2: $j \leftarrow random(1, m)$
3: **if** $X_j \in \mathbf{X}'$ **then**
4: $\mathbf{X}' \leftarrow \mathbf{X}' \setminus X_j$
5: **else**
6: $\mathbf{X}' \leftarrow \mathbf{X}' \cup X_j$
7: **end if**
8: **end for**

The local search proposed in this work, called Line Search, is intended to fit the parameters of the model. These parameters belong to the real numbers domain (i.e., $\{\epsilon, \alpha_i, \beta_i\}, \in \mathbb{R}\, \forall i \in m'$). Therefore, the parameter adjustment is suitable to be solved by global optimization strategies. Line Search is one of the most commonly used strategies for improving solutions in global optimization, which has lead to several successful research [28]. Given a solution $S = (\mathbf{X}', \mathbf{W})$ and a parameter $p_i \in \{\epsilon, \alpha_i, \beta_i \forall i \in |\mathbf{X}'|\}$ of the model, the Line Search for S in the p_i direction is represented as $ls(S, h, p_i)$, where h is the width of the uniform grid of the discretized search space. Therefore, it is possible to reach all the feasible solutions belonging to the h-grid by modifying parameter p_i in S. Algorithm 3 presents the pseudo-code of the line search method proposed.

Algorithm 3. LineSearch(S, h)

1: $\mathbf{C} \leftarrow \{1, 2, \ldots, 2m' + 1\}$
2: **while** $|\mathbf{C}| > 0$ **do**
3: $j \leftarrow random(\mathbf{C})$
4: $\mathbf{C} \leftarrow \mathbf{C} \setminus \{j\}$
5: $w \leftarrow LB_j$
6: **while** $w \leqslant UB_j$ **do**
7: $S' \leftarrow UpdateSolution(S, j, w)$
8: **if** $f(S') < f(S)$ **then**
9: $S \leftarrow S'$
10: $\mathbf{C} \leftarrow \{1, 2, \ldots, 2m' + 1\}$
11: **end if**
12: $w \leftarrow w + (UB_j - LB_j)/h$
13: **end while**
14: **end while**

The method starts by creating a list with all the parameters involved in the search (step 1). Then, it iterates over the parameter list until there is not available parameters to improve (steps 2–14). In each iteration, the method selects a parameter at random from the list of available parameters (step 3), removing it from the list (step 4). Then, given the lower (LB_i) and upper (UB_i) bounds of the parameter (given by the model), the method traverses the grid for this parameter, starting in the lower bound (step 5), and performing a total of h evaluations in the grid of the parameter, where h is a parameter of the algorithm. For each iteration, the method evaluates the quality of the solution when setting the new value of p_i (step 7). If an improvement is found, the solution is updated and the list of parameters to be explored is reset to the initial set to restart the search (steps 9–10).

3.2. Extreme Learning Machines

The second part of the prediction is performed by an Extreme Learning Machine (ELM), which is based on the structure of Multi-Layer Perceptrons (MLP) but with a extremely fast training stage. ELM was recently proposed in [29], and it has been successfully applied to a wide variety of classification and regression problems [30–32]. The structure of the ELM used in this problem is similar to the one given in Fig. 1. The ELM consist of three main layers: the input layer, which is intended to receive the set of inputs to be used; the hidden layer, where the inputs are evaluated using a given activation function and calculating the weights associated to them; and, finally, the output layer, where the ELM returns the value obtained for a given input.

The most relevant feature of ELM is the reduced computing time needed for the training stage. This is mainly because the training mostly consists of randomly setting the network weights and then computing the inverse of the hidden layer output matrix. Furthermore, this faster training stage does not affect to the performance of the ELM when comparing it with traditional neural

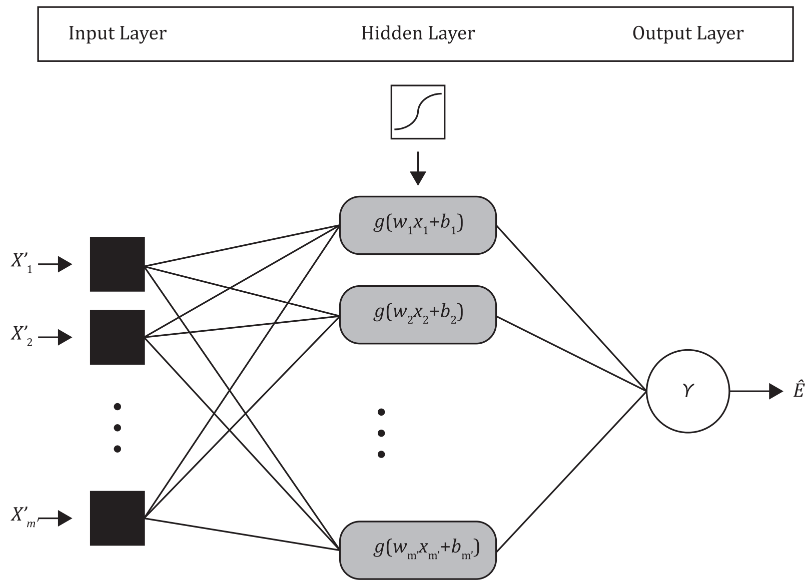

Fig. 1. Outline of the Extreme Learning Machine structure.

networks, resulting in better outcomes than classical MLPs or Support Vector Machines, see [33] for a deeper analysis in ELM theory.

In mathematical terms, the ELM consists of a training set $\mathcal{N} \triangleq \{(x_i, t_i) | x_i, t_i \in \mathbb{R}^n, i = 1, \ldots, N_T\}$ (where N_T represents the number of samples selected for the training set), an activation function $g(x)$ and a number of hidden nodes $\widetilde{N}$. The training stage of the ELM can be divided in three steps. In the first one, it assigns at random input weights w_i and bias b_i for each $i \in \widetilde{N}$. The second step consists of calculating the output matrix H for the hidden layer, using the activation function and the weights of the first step. Specifically, the element (i,j) of h is given by $g(w_i x_j + b_i)$. Finally, the third step calculates the output weight vector γ as the Moore-Penrose inverse of H [29] multiplied by the training output vector.

The ELM needs two additional parameters to work: the number of hidden nodes and the activation function. While the former is a free parameter and needs to be adjusted in order to obtain good results (normally it is performed by scanning a range of integer values), the latter is more limited, to a number of well-known activation functions: sigmoidal, sine, hardlim, etc. Then, it is necessary to perform a preliminary test to select the best number of hidden nodes and the best activation function according to the problem under consideration.

3.3. Hybrid approach

The algorithm proposed consists of a hybrid approach which combines the Basic Variable Neighborhood Search with an Extreme Learning Machine in order to perform the energy prediction. The algorithm uses BVNS to select the best macroeconomic features from the set of available ones, while the aim of ELM is to perform the energy demand prediction based on the selected features.

Fig. 2 represents the flowchart of the BVNS+ELM algorithm. It starts by executing the BVNS method during a maximum number of iterations. The initial solution for BVNS is a random selection of features with their associated weights also randomly selected, $\{X^{rnd}, W^{rnd}\}$. Then, BVNS is executed a fixed number of iterations, starting each of them from the best solution found during the search, $\{X^{bvns}, W^{bvns}\}$. BVNS finishes when the maximum number of iterations is reached. ELM then performs the training phase using the features selected by BVNS, X^{bvns}, being able to perform an estimation of the energy demand $\hat{E}$ with the parameters adjusted for the previously selected features.

4. Experimental part

The test case considered for the experimentation[1] is a real problem of energy demand estimation in Spain, previously addressed in [18]. The available data have been collected in the year range 1980 to 2011. For each year, a set of $m = 14$ variables have been recovered. Specifically,

1. Gross Domestic Product (GDP).
2. Population.
3. Exports (amount of money traded in exporting goods in Euros).
4. Imports (amount of money traded in importing goods in Euros).
5. Energy production (kTOE).
6. Electricity power transport (kW/h).
7. Electricity production (kW/h).
8. GDP per unit of energy use.
9. Energy imports net (% of use).
10. Fossil fuel consumption (kW/h).
11. Electric power consumption (kW/h).
12. CO_2 emissions total (Mtons).
13. Unemployment rate.
14. Diesel consumption in road (kTOE).

The data has been split into training and test sets. The former consist of data obtained in 15 different years, while the latter contains the data of the remaining 16 years. Training and test sets have been generated at random, but keeping years 2010 and 2011 in the test set in order to check the behavior of the model when considering years of crisis.

The main goal of the proposed algorithm is to become a useful tool to facilitate the work of a decision maker. Considering that the BVNS (devoted to select the best subset of features) and the ELM training (devoted to optimize the energy prediction demand) are

[1] This data can be downloaded at http://www.optsicom.es/edpp/eddp.zip.

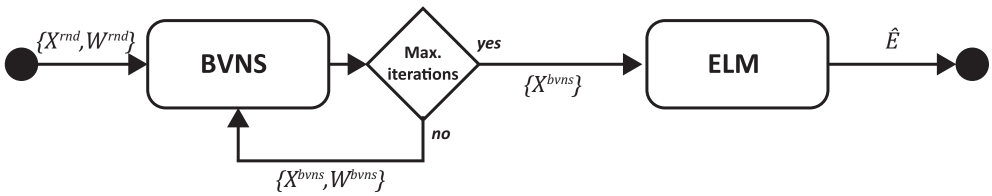

Fig. 2. Flowchart of the hybrid BVNS+ELM algorithm.

executed off-line (i.e., they are run only once), computing time is not a relevant result to be analyzed. In other words, a decision maker only needs to feed the BVNS+ELM algorithm (already optimized) with a given subset of features and it returns almost immediately the corresponding energy demand prediction. However, for the sake of completeness, note that the computing time for the BVNS is set to 10 s per iteration, yielding to maximum of 5 min, while the ELM requires about 5 s to optimize the prediction.

The first experiment is intended to select the best parameters for the BVNS algorithm. A value of 1000 has been experimentally set for the Line Search parameter h in all variants. In this case, the number of iterations has been set to 25 (in order to maintain a reasonable computing time) and different values for k_{max} has been tested (specifically, $k_{max} = \{2, 4, 6, 10\}$). Table 1 shows the performance of each variant, reporting the relative mean absolute error (MAE, in %) obtained in the test set once the exponential parameters have been fixed.

The results presented in Table 1 show that $k_{max} = 4$ is the best variant, resulting in a MAE of 2.10%. It is important to remark that the MAE increases together with the size of the largest neighborhood. However, the smallest neighborhood ($k_{max} = 2$) presents the worst results with respect to MAE, which may indicates that $k_{max} = 2$ is a relatively small neighborhood to be used in this problem. These results agree with the observation made in [23], where authors recommend to use small values of k_{max}. Therefore, it has been decided to use $k_{max} = 4$ in the BVNS algorithm.

Fig. 3 shows the comparison between the energy demand prediction obtained with BVNS and the real values. As it can be seen, the prediction is really accurate, even in the years of the economic crisis, starting in 2008. Regarding the selected variables, it is important to remark that macroeconomic variables import (4), energy imports net (9), and electric power consumption (11), have been selected in the four variants. The best one has additionally selected GDP per unit of energy use (8) and unemployment rate (13), resulting in a subset of six different variables selected.

The main objective of the next experiment is to select the best activation function and number of neurons in the hidden layer for the proposed ELM. A set of traditional activation functions has been selected: sigmoid (sig), sine (sin), hard limit (hardlim), triangular basis (tribas), and radial basis (radbas). The number of neurons in the hidden layer, # Neurons, is selected by scanning the integer range [5, 10]. Table 2 presents the performance of the ELM with all the possible combinations of the selected parameters. Specifically, for each pair of activation function and number of neurons (# Neurons), the minimum (min) and the average (avg) MAE obtained in 10 runs over the training set have been reported.

These results allow us extracting relevant information about the ELM configuration. In this case, the most remarkable idea that can be derived is that the results obtained by the hard limit function are not competitive with the remaining results. This can be partially explained by the binary nature of the function, with no smoothness in the weight evaluation. In the case of triangular basis, the results, although better than hard limit, are not good enough to be considered on average when compared with radial basis, sine or sigmoid.

Table 1
Performance of BVNS with different values of k_{max}. Common features selected in all cases are highlighted in bold font.

k_{max}	MAE (%)	Selected variables
2	3.40	{**4**, 7, **9**, **11**, 13, 14}
4	2.10	{3, **4**, 8, **9**, **11**, 13}
6	2.59	{1, 3, **4**, 5, 8, **9**, **11**}
10	2.86	{2, 3, **4**, 6, 7, **9**, **11**, 12}

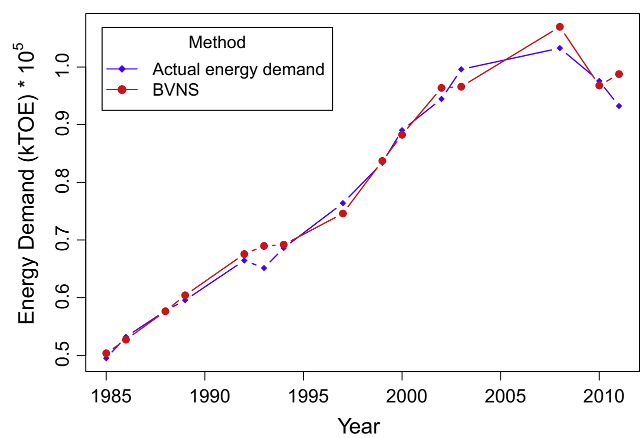

Fig. 3. Real energy demand versus BVNS prediction in the test set considered.

If the sine function is analyzed, it can be seen that it maintains an average good performance in all the cases, but it is not able to outperform the results of the sigmoid or sine function, which are able to obtain the best results when considering all possible number of neurons in the hidden layer. Both sigmoid and radial basis present a comparable performance, but in this case, sigmoid has been selected as activation function, and 7 neurons in the hidden layer for the final experiment, since this configuration presents the lowest value in both minimum and average MAE.

The final experiment consists of analyzing the performance of the ELM once the BVNS has already selected the best features for the exponential model. Therefore, the ELM has been trained with the features selected by BVNS presented in Table 1: {3, 4, 8, 9, 11, 13}. Since the ELM is not deterministic, it has been executed 10 times, storing the best MAE obtained. The obtained results are compared with the ELM when no feature selection has been performed (i.e., it uses all the available features). In order to better situate the contribution of the proposed procedure, it has been also compared with the current state of the art. Specifically, the results obtained by a Harmony Search (HS) approach (that uses the same exponential model), and an ELM which uses the features previously selected by the HS, have been included. The reader is referred to [18] for thorough description of these approaches.

Table 3 shows the results obtained. The first and second rows report the performance of the heuristic procedures to select features. The proposed BVNS achieves a remarkable MAE (2.10%)

Table 2
Performance of ELM with different activation functions and number of neurons in the hidden layer.

		Activation function									
		sig		sin		hardlim		tribas		radbas	
		min	avg	min	avg	min	avg	min	avg	min	avg
# Neurons	5	2.21	4.56	2.07	3.41	10.36	20.11	3.41	9.54	1.66	3.40
	6	1.44	2.91	1.18	2.31	15.70	20.59	1.80	6.18	1.50	2.15
	7	**0.36**	**0.90**	1.02	1.76	11.38	21.37	2.00	9.63	0.94	1.62
	8	1.04	1.42	0.90	1.32	6.08	18.44	1.73	2.77	0.95	1.58
	9	0.76	1.16	0.56	0.97	9.93	16.86	0.86	2.53	0.38	0.96
	10	0.48	0.92	0.51	0.94	9.58	14.85	0.69	1.40	0.63	0.91

Bold values stand for the best case found in the experiments.

Table 3
Performance of BVNS with different values of k_{max}.

k_{max}	Best MAE (%)	Average MAE (%)	Selected variables
HS	2.36	4.05	[1, 2, 3, 7, 8, 9, 12]
BVNS	2.10	–	[3, 4, 8, 9, 11, 13]
ELM	2.22	4.16	All
HS+ELM	2.16	3.72	[1,2,3,7,8,9,12]
BVNS+ELM	**1.66**	3.90	[3, 4, 8, 9, 11, 13]

Bold value stands for the best case found in the experiments.

while the HS obtains a 2.36% (in the best case) and 4.05 % (in average). Notice that BVNS is a deterministic algorithm, so the best and average results are exactly the same. The best solution obtained with the BVNS algorithm is the following:

$$\hat{E} = 0.69 + 0.28 \cdot X_3^{0.08} - 0.026 \cdot X_4^{-0.996} - 0.142 \cdot x_8^{0.44} + 0.004 \cdot X_9^{-0.71} + \cdots + 0.778 \cdot X_{11}^{0.872} + 0.02 \cdot X_{13}^{-0.692} \quad (3)$$

The selected features can be used by an alternative regression method to further improve the results. Specifically, an ELM has been chosen, which has shown an excellent performance in previous regression problems [29]. The performance of the ELM that considers the whole set of features has been set as the baseline approach to be beaten. It obtains a best and average MAE of 2.22% and 4.16%, respectively. The proposed BVNS+ELM is able to outperform the best procedure in the related literature (HS+ELM), when considering the best MAE (1.66% vs. 2.16%). Notice that the HS+ELM presents a slightly better behavior in terms of average MAE. This fact can be partially explained by the stochastic nature of the ELM.

Fig. 4 represents the prediction performed by BVNS, ELM and ELM with BVNS feature selection (BVNS+ELM) when compared with the actual energy demand for each year selected in the test set. As it can be seen, the three proposed algorithms fits good with the actual values. However, the combination of BVNS with ELM is able to obtain a better fit in the conflictive years related to the crisis, and a special case, 1994, where there was a decrement in the energy demand. For that reasons, the combined method emerges as the best algorithm to predict the energy demand.

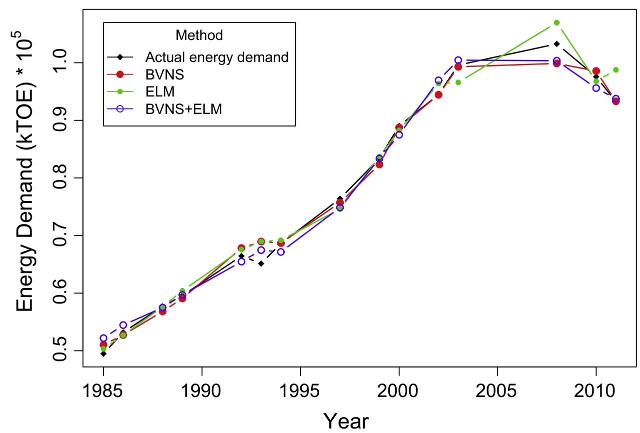

Fig. 4. Real energy demand versus BVNS, ELM and BVNS+ELM prediction in the test set considered.

5. Conclusions

This paper presents a novel hybrid meta-heuristic approach formed by a Variable Neighborhood Search (VNS) and an Extreme Learning Machine (ELM) neural network for tackling a problem of total energy demand estimation in Spain. VNS is a meta-heuristic that has shown a great potential to explore difficult search spaces. It results as an excellent global search approach in this problem of energy demand estimation, where it is used to obtain a reduced set of features for tackling the prediction of the energy demand. On the other hand, the ELM is an extremely fast training neural network, recently proposed, and applied to a large variety of different prediction problems. The hybridization of both techniques leads to a robust approach to tackle the prediction of the total energy demand from socio-economic variables. Experiments in a real case in Spain have shown the good performance of the proposed hybrid approach, by obtaining better results than previous approaches for this problem. The final prediction system designed is able to obtain accurate one-year-ahead energy demand estimation, which works even in the crisis years (from 2008 and years after). This system could be of help for policy makers and practitioners who must deal with energy demand estimation at nation level.

Acknowledgements

This work has been partially supported by the projects TIN2014-54583-C2-2-R and TIN2015-65460-C2-2-P of the Spanish Ministerial Commission of Science and Technology (MICYT), and by the Comunidad Autónoma de Madrid, under project numbers S2013ICE-2933_02 and S2013ICE-2894.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.enconman.2016.06.050.

References

[1] Salisu AA, Ayinde TO. Modeling energy demand: some emerging issues. Renew Sustain Energy Rev 2016;54:1470–80.
[2] Suganthi L, Samuel AA. Energy models for demand forecasting – a review. Renew Sustain Energy Rev 2012;16:1223–40.

[3] Ceylan H, Ozturk HK. Estimating energy demand of turkey based on economic indicators using genetic algorithm approach. Energy Convers Manage 2004;45:2525–37.
[4] Ediger V, Akar S. Arima forecasting of primary energy demand by fuel in turkey. Energy Policy 2007;35:1701–8.
[5] Toksari MD. Ant colony optimization approach to estimate energy demand of turkey. Energy Policy 2007;35:3984–90.
[6] Toksari MD. Estimating the net electricity energy generation and demand using the ant colony optimization approach: case of turkey. Energy Policy 2009;37:1181–7.
[7] Ünler A. Improvement of energy demand forecasts using swarm intelligence: the case of turkey with projections to 2025. Energy Policy 2008;36:1937–44.
[8] Kiran MS, Özceylan E, Gündüz M, Paksoy T. Swarm intelligence approaches to estimate electricity energy demand in turkey. Knowl-Based Syst 2012;36: 93–103.
[9] Kiran MS, Özceylan E, Gündüz M, Paksoy T. A novel hybrid approach based on particle swarm optimization and ant colony optimization to forecast energy demand of turkey. Energy Convers Manage 2012;53:75–83.
[10] Kankal M, Akpinar A, Kömürcü MI, Özsahin TS. Modeling and forecasting of turkey's energy consumption using socio-economic and demographic variables. Appl Energy 2011;88:1927–39.
[11] Adnan S. Universal approximation using incremental constructive feedforward networks with random hidden nodes. Energy Policy 2009;37:4827–33.
[12] Geem ZW, Roper WE. Energy demand estimation of South Korea using artificial neural network. Energy Policy 2009;37:4049–54.
[13] Ekonomou L. Greek long-term energy consumption prediction using artificial neural networks. Energy 2010;35:512–7.
[14] Yu S, Zhu KJ. A hybrid procedure for energy demand forecasting in China. Energy 2012;37:396–404.
[15] Yu S, Zhu K, Zhang X. Energy demand projection of China using a path-coefficient analysis and PSO-GA approach. Energy Convers Manage 2012 (53):142–53.
[16] Yu S, Wei YM, Wang K. A PSO-GA optimal model to estimate primary energy demand of china. Energy Policy 2012;42:329–40.
[17] Piltan M, Shiri H, Ghaderi SF. Energy demand forecasting in iranian metal industry using linear and nonlinear models based on evolutionary algorithms. Energy Convers Manage 2012;58:1–9.
[18] Salcedo-Sanz S, Muñoz-Bulnes J, Portilla-Figueras J, Ser JD. One-year-ahead energy demand estimation from macroeconomic variables using computational intelligence algorithms. Energy Convers Manage 2015;99:62–71.
[19] Bianco V, Manca O, Nardini S. Electricity consumption forecasting in Italy using linear regression models. Energy 2009;34:1413–21.
[20] Mohamed Z, Bodger P. Forecasting electricity consumption in New Zealand using economic and demographic variables. Energy 2005;30:1833–43.
[21] Blum A, Langley P. Selection of relevant features and examples in machine learning. Artif Intell 1997;97:245–71.
[22] Mladenović N, Hansen P. Variable neighborhood search. Comput Oper Res 1997;24:1097–100.
[23] Hansen P, Mladenović N, Moreno-Pérez J. Variable neighborhood search: methods and applications. Ann. Oper. Res. 2010;175(1):367–407.
[24] Pardo E, Mladenović N, Pantrigo J, Duarte A. Variable formulation search for the cutwidth minimization problem. Appl Soft Comput 2013;13(5):2242–52.
[25] Sánchez-Oro J, Mladenović N, Duarte A. General variable neighborhood search for computing graph separators, Optim Lett. doi:http://dx.doi.org/10.1007/s11590-014-0793-z.
[26] Sánchez-Oro J, Pantrigo J, Duarte A. Combining intensification and diversification strategies in VNS. An application to the Vertex Separation Problem. Comput Oper Res 2014;52(B):209–19.
[27] Selvakumar AI, Thanushkodi K. A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE Trans Power Syst 2007;22: 42–51.
[28] Duarte A, Martí R. Hybrid scatter tabu search for unconstrained global optimization. Ann Oper Res 2011;183(1):95–123.
[29] Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing 2006;70(1):489–501.
[30] Huang GB, Wang DH, Lan Y. Extreme learning machines: a survey. Int J Mach Learn Cybernet 2011;2(2):107–22.
[31] Huang G-B, Chen L. Convex incremental extreme learning machine. Neurocomputing 2007;70(16–18).
[32] Huang G-B, Chen L. Enhanced random search based incremental extreme learning machine. Neurocomputing 2008;71(16–18):3460–8.
[33] Huang G-B, Chen L, Siew C-K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Networks 2006;17(4):879–92.