Martin, R., Aler, R., Valls, J. M., and Galvan, I. M. (2016) Machine learning techniques for daily solar energy prediction and interpolation using numerical weather models. Concurrency Computat.: Pract. Exper., 28: 1261– 1274.

DOI: 10.1002/cpe.3631

CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2016
Published in Wiley Online Library (wileyonlinelibrary.com).

SPECIAL ISSUE PAPER

Machine learning techniques for daily solar energy prediction and interpolation using numerical weather models

R. Martin, R. Aler*,†, J. M. Valls and I. M. Galvan

Computer Science Department, Carlos III University of Madrid, 28911 Leganés, Madrid, Spain

SUMMARY

This article addresses two issues in solar energy forecasting from the numerical weather prediction (NWP) models using machine learning. First, we are interested in determining the relevant information for the forecasting task. With this purpose, a study has been carried out to evaluate the influence on accuracy of the number of NWP grid nodes used as input for the forecasting model, as well as their relative importance. Several machine learning (support vector machines and gradient boosting) and feature selection algorithms (linear, ReliefF, and local information analysis) have been used in this study. The second aim is to be able to predict solar energy for locations where no previous production data are available. To address this goal, an approach consisting on modeling regions in the grid is proposed. Models (aggregate models) use as input attributes the meteorological variables relevant for the region and two new inputs to identify the location of each station: the latitude and the longitude. Those models can be used to predict energy production for existing stations and for new locations, represented by latitude and longitude. Copyright © 2015 John Wiley & Sons, Ltd.

Received...

KEY WORDS: forecasting solar energy, solar energy interpolation, machine learning methods

1. INTRODUCTION

Photovoltaic systems are becoming important sources of energy in electricity networks. However, electric utility companies are required to guarantee electricity supply within certain ranges in future periods, which is difficult given the fluctuating nature of weather conditions. Thus, accurate forecast of solar radiation is becoming an important issue in the context of renewable energy sources. An approach to forecasting is to use statistical and machine learning techniques based on historical data of solar production [1]. With respect to the machine learning techniques using historical data, many works appear in the literature that use, for instance, artificial neural networks [2] or support vector machines (SVM) [3, 4]. However, for the prediction horizons required by photovoltaic plants (day-ahead), it has been shown that models based on numerical weather prediction (NWP) systems, such as the global forecast system and the European Center for Medium-Range Weather Forecast, are a good alternative [1]. These models forecast meteorological variables for points in a low-resolution grid. NWP-forecasted variables have been used as input for machine learning techniques mainly for wind power prediction [5, 6] and recently for solar energy forecasting [7, 8].

Here, we are interested on issues related to predicting incoming solar energy from NWP models computed from the National Oceanic and Atmospheric Administration/Earth System Research Laboratory Global Ensemble Forecast System (GEFS). GEFS provides short-term forecasting for

*Correspondence to: R. Aler, Computer Science Department, Carlos III University of Madrid, Avda. de la Universidad N° 30, 28911 Leganés, Madrid, Spain.

†E-mail: aler@inf.uc3m.es

several meteorological variables, for different points or nodes located in a grid. In principle, the meteorological variables of the closest grid nodes to the solar station should be the most relevant for prediction, but it can be useful to know how prediction accuracy improves as more and more GEFS grid nodes are used as input for the machine learning techniques. Therefore, the first goal of this paper, initially addressed in [9], is to study the influence of the number of grid nodes on the prediction accuracy. This involves building individual models for places where historical solar radiation data are available. These individual or local models can take into account the characteristics of each solar site, which may be different because of the topography of the area where it is located, microclimate, and so on. However, these models can only predict energy production for that particular location. Thus, the second goal addresses solar radiation prediction for new solar stations that could be deployed on a new location in the future, where no such historical data are known.

For this paper, we use the data supplied by Kaggle[‡] where the goal was to predict the total daily solar energy at 98 Oklahoma mesonet solar sites using 15 NWP variables every 3 h spread on a 16×9 spatial grid. With respect to the first issue addressed in this article, two state-of-the-art machine learning techniques (SVM [10] and gradient boosted regression (GBR) [11, 12]) have been used to build models for each solar station, varying the number of GEFS grid nodes used as input for the models, starting from the closest grid nodes. For the cases where many grid nodes are used, the resulting large number of attributes might worsen the generalization capabilities of the learning algorithms. Therefore, feature selection methods have been applied to study whether prediction accuracy could be improved using a selection of input attributes. In the literature, a large variety of feature selection methods can be found [13, 14]. In this work, three filter methods are used: linear correlation (one of the simplest methods), ReliefF algorithm (a commonly used robust algorithm) [15], and the local information analysis (LIA) algorithm (an advanced and fast method that uses local reliable information provided by each attribute) [9].

To address the second goal (prediction in new locations), a direct approach would be to build a global model using the available data from all the 98 stations and adding two new inputs to the model in order to identify the location of each station: the latitude and the longitude. Thus, the model would be trained with the whole data, and it could be applied to any location (in terms of latitude and longitude), not only to the 98 stations given in the problem. However, training such a global model in the context of this problem is computationally very expensive. An alternative option is to split the computation of the global model, building specific models for localized regions in the grid and grouping the subsets of the 98 stations included in the region. More specifically, the proposed approach is to learn one model for every square region in the grid, grouping solar sites within each square. Those models, which we have called aggregate models, use the information of all the solar stations within the square region. In addition to the NWP variables of the surrounding grid nodes (which were already used for the individual models), the latitude and longitude are added as input variables in order to provide information to the model about the localization of the sites.

Some works found in the literature address the problem of estimating meteorological variables in new locations. For instance, in [16], different methods, such as inverse distance weighted, natural neighbor, and kriging [17] (a statistical method), are compared with estimate precipitation, temperature, and wind speed in unsampled locations. Kriging methods together with orographic corrections are also used in [18] to interpolate wind speed. In the same line, Alsamamra *et al.* [19] use a kriging method with topographic external variables to estimate global solar radiation in new locations. As a final example, in [20], a neural network is used as the method for interpolating solar radiation, instead of kriging, by using latitude and longitude as input to the network. The aim of these works is to use known measurements of meteorological variables to compute the same variable at unsampled locations. However, although our goal is also to estimate a variable (solar energy) at new locations, there are two main differences: (1) the inputs to our model are not measurements of the same variable (solar radiation in this case) but the surrounding NWP grid nodes and (2) the final model can be used to predict solar radiation in the future, rather than predicting the variable at the same time the measurements have been taken.

[‡]https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest.

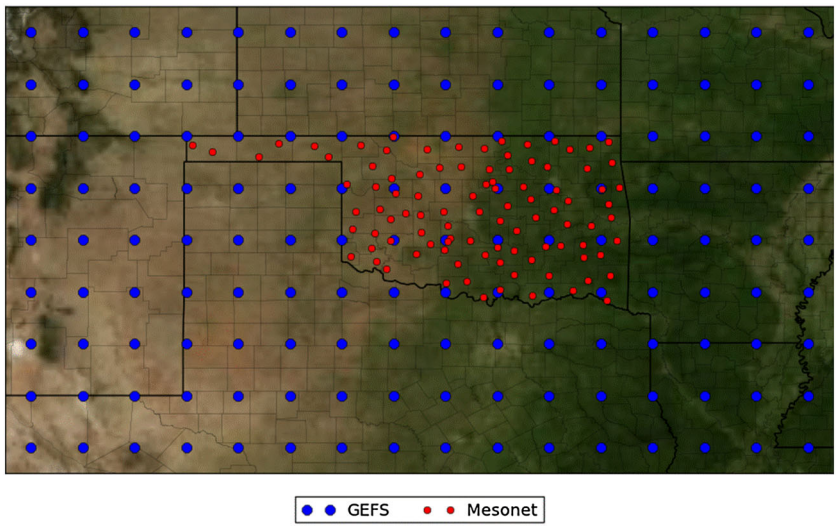

Figure 1. Global Ensemble Forecast System (GEFS) grid points and mesonet stations.

The rest of the paper is structured as follows: Section 2 describes the data used in this work to address the two stated goals. Section 3 explains all machine learning methods used in this research. This section includes the feature selection methods involved in the first part and the regression methods used in the first and second parts. The first goal of the paper is dealt with in Section 4 (Part I). This section presents the results of the performance of the different machine learning techniques as the number of grid nodes is increased and the results of the study of different feature selection methods for solar energy production forecasting. Section 5 (Part II) describes the aggregate models, which is the approach for solar energy prediction at new locations, explaining the procedure to train and validate those models. This section includes the experimental results, both for existing and new locations. Finally, Section 6 provides the conclusions of this work.

2. DESCRIPTION OF DATA

The data available from the Kaggle website have been provided by the American Meteorological Society.[§] The goal is to predict the total daily incoming solar energy, measured in $J \times m^{-2}$, at 98 sites of the Oklahoma Mesonet network, which covers a surface of, approximately, 180,000 km^2. The input data for each day correspond to the output of the NWP model GEFS using 11 ensemble members and five forecast timesteps from 12 to 24 h in 3-h increments. Each ensemble member produces outputs for 15 different meteorological variables for each timestep and each point of a 16×9 uniform land-surface grid, with a spatial resolution of about 90 km, including the State of Oklahoma and surrounding areas. Figure 1 shows the GEFS grid nodes (blue points) and the 98 Oklahoma Mesonet sites (red points). Some of the meteorological variables used are the following: accumulated precipitation ($kg\ m^{-2}$), air pressure (Pa), downward and upward short-wave/long-wave radiation ($W\ m^{-2}$), cloud cover (%), and temperature (K). A more detailed information can be found in.[§] Thus, the number of attributes for each grid node is $11 \times 5 \times 15 = 875$. Because the number of grid points is $16 \times 9 = 144$, the total amount of available data for each day equals 118,800. Data have been collected everyday from 1994 to 2007 (5113 days) in association with the corresponding accumulated incoming solar energy, which is the attribute to be predicted. This accumulated incoming solar energy (in $J \times m^{-2}$) has been calculated by summing the solar energy measured by a pyranometer at each mesonet site every 5 min, from the sunrise to 23:55 universal time coordinated of the corresponding date.

[§]https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest.

From the total input–output available data covering 14 years, we have used the period 1994–2005 as the training set (4383 days), reserving the period 2006–2007 (730 days) for the testing set.

3. METHODS

3.1. Regression methods

The machine learning techniques tested in this work to predict the energy production have been SVM and GBR.

Support vector machines [10] are a class of supervised learning method extensively applied to classification and regression problems. SVMs basically construct maximum margin hyperplanes and use kernel functions to build non-linear models. The most commonly used kernel functions are linear, polynomial, and the radial basis function (RBF) kernels. Accuracy is greatly influenced by the cost parameter C and the kernel parameters (σ in the case of the most commonly used kernel, the RBF). A more detailed information about SVMs can be found in [21, 22]. In this work, we have tested both linear and RBF kernels, and we have used the Waikato Environment for Knowledge Analysis SVM implementation called sequential minimal optimization (SMO) [23].

Gradient boosted regression is a recent machine learning technique that has shown considerable success in predictive accuracy. The method was proposed by Friedman [11, 12], and it produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. GBR uses two algorithms: regression trees and boosting (an adaptive method for combining many simple models to give improved predictive performance) that builds and combines a collection of models (regression trees, in this case). Boosting constructs ensembles by sequentially adding models, so that each model focuses on mistakes made by the previous model. Like SVM, the accuracy of GBR models depends on some parameters, as the number of trees used, the shrinkage (a regularization parameter) and the depth of trees. A more detailed description can be found in [24]. We have used for experiments the gbm package [25] from the R language [26].

3.2. Attribute selection methods

In this work, three feature weighting attribute selection algorithms have been used: linear correlation, ReliefF, and LIA. The linear correlation attribute selection method ranks attributes according to their linear correlation with the target. The ReliefF algorithm [15] estimates the quality of attributes in problems with strong dependencies between attributes. The estimation of the quality of attributes is made according to how well their values distinguish between instances that are near to each other. It evaluates an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. In our work, the algorithm implementation for Weka tool [23] (ReliefFAttributeEval) has been used as the attribute evaluator and Ranker as search method. The LIA algorithm uses local information to evaluate the relevance of the attributes, and it is described in more detail as follows.

Attribute selection algorithm based on local information analysis. Basically, the algorithm consists of evaluating all possible subsets 1 and 2 attributes, mapping each input datum into grids of dimensions 1 and 2, respectively, using the value of the corresponding attributes. The algorithm assumes that the output is binary; that is, patterns belong to two classes: $C0$ and $C1$. For regression, as in this work, the problem is transformed into 10 binary problems, by discretizing the output value in 10 intervals. The attribute selection algorithm is applied to each problem, and the rankings of the 10 sets of attributes are combined.

The algorithm is based on computing the reliable information provided by each attribute in the target class, by means of an evaluation function F_I that estimates the number of instances belonging to the target class that are not due to randomness (given a confidence value), within a cell of the grid. In order to define F_I, the following notation will be used:

- C_1 and C_0: C_1 is the target class. C_0 is the alternative class.
- $Total.C_1$ and $Total.C_0$: Number of C_1 and C_0 patterns in the dataset.

- *Cell*: A cell in the grid
  - $Cell.C_1$ and $Cell.C_0$: Number of C_1 and C_0 patterns in the cell.
  - $Cell.Total = Cell.C_0 + Cell.C_1$
- *Confidence* value[¶]
- $Density = \frac{Total.C_1}{Total.C_1 + Total.C_0}$.

Given that this technique has been designed for binary problems, the statistical tool for modeling probabilities is the binomial distribution. Therefore, the function used to estimate randomness effects in a *Cell*, for a given *Confidence* and a *Density* (determined by the problem) is given by Equation 1.

$$F_E(Cell, Confidence, Density) = IBD(Cell.Total, Confidence, Density) \tag{1}$$

where *IBD* is the inverse of the binomial distribution, that is, a function that returns the number of events x such as the cumulative probability $p(X \leqslant x) \leqslant Confidence$ in a binomial distribution with parameters $n = Cell.Total$ and $p = Density$. Therefore, F_E returns a number in $[0, N]$, where N is any positive number.

Equation (2) displays the F_I function, which computes how many C_1 instances within a cell cannot be explained by chance alone, and returns a value within $[0, Cell.C_1]$. In order to formalize F_I function, a parameter *CP* (named confidence parameter) is defined to determine the value of *Confidence* that is used to compute the information provided by each attribute. Basically, $Confidence = 1 - 10^{-CP}$.

$$F_I(Cell, CP) = \max((Cell.C_1 - F_E(Cell, Confidence, Density), 0.0)) \tag{2}$$

Next, the full attribute selection algorithm will be described in detail. Previously, some notations and parameters are introduced:

- Grids with P^{dim} cells are constructed by dividing $[0, 1]$ in P parts (in this work $P = 4$). *dim* is the dimension of the grid. In this study $dim = 1$ and $dim = 2$
- *VR* is a vector with as many components as attributes (*nAtrs*). It will be used to store the weight of each attribute for dimension *dim*.
- *MI* is a matrix of dimension $nAtrs \times C$. It will be used to accumulate the information provided by each attribute, for every confidence parameter *CP* used.
- *VRT* is a vector with as many components as attributes (*nAtrs*). It will be used to store the total weight of each attribute.
- The confidence parameter *CP* takes values from $CP_{\min}$ to $CP_{\max}$ by steps of size Δ. All possible values of *CP* are stored in the *VConf* vector: $VConf_i = CP_{\min} + i * \Delta$ for $i = 0 \ldots C$ where $C = \frac{CP_{\max} - CP_{\min}}{\Delta}$. In the experiments of this paper, $CP_{\min} = 0.315$, $CP_{\max} = 7.305$, and $\Delta = 0.25$. This corresponds, respectively, to levels of $Confidence_{\min} = 0.5$ and $Confidence_{\max} = 0.9999999$.

The steps of the algorithm are the following:

1. *VRT* is initialized to zero.
2. Grids with dimension 1 are processed ($dim = 1$).
3. *VR* and *MI* are initialized to zero.
4. For each combination of *dim* attributes (in the case of *dim > 1* only are computed those combinations of attributes in which there is an attribute that has provided information when grids of dimension $dim = 1$):

[¶]This parameter is not exactly equivalent to statistical confidence because there are other factors that must be taken into account, such as data distribution and autocorrelation.

- The grid with P^{dim} cells is constructed. Each training pattern is mapped to its corresponding cell, according to the attributes considered in the grid. The number of patterns $Cell.C_1$ and $Cell.C_0$ is computed for every cell.
- F_I (Equation 2) is used to compute the information provided in each cell and for every confidence parameter in *VConf*, and values obtained are accumulated in *MI* as follows:

$$MI_{ij} += \sum_{Cell} F_I\left(Cell, VConf_j\right) j = 1 \ldots C, i \in \{ProcesedAttributes\} \quad (3)$$

5. In order to assign a weight to every attribute, *MI* is processed starting from the last column (corresponding to the maximum confidence $Confidence_{\max}$ or $CP_{\max}$) and decreasing the confidence level until all attributes (or the desired number of attributes) have been assigned a weight. The sequence of steps is as follows:
   - $k = C$.
   - The attribute i with maximum information in MI_{ik} and $VR_i = 0$ is assigned a weight *nAtrs*. The next attribute is assigned a weight $nAtrs - 1$, and so on. This is repeated for all attributes that have information for that confidence level. These weights are stored in vector *VR*.
   - If the number of desired attributes max $Atrs$ has not been reached and $k > 1$, then the confidence level is decreased, and therefore, column $k = k - 1$ is processed. Then, the previous step is repeated.
6. $VRT_i = VRT_i + VR_i, i \in \{1 \ldots nAtrs\}$
7. If $dim < 2$, then $dim = dim + 1$ and the algorithm continues at step 2.
8. Attributes are ranked according to *VRT* in order descendent. The first is the most relevant, and so on.

3.3. A note on parallelization

In the problem addressed in this work, notwithstanding whether individual (Part I) or aggregate models (Part II) are learned, a large amount of them has to be constructed, and this can be carried out in parallel. For instance, given that there are 98 solar stations, 98 individual models have to be estimated. Even when dealing with aggregate models, there are 24 of them. Also, interpolation performance is measured by means of cross-validation that multiplies the number of models to be constructed. Finally, optimal parameter adjustment of the learning algorithms by means of exhaustive grid search also offers many opportunities for straightforward parallelization. To summarize, in this work, each different model (of the many to be estimated) is learned in parallel, in a different CPU or core.

There are many possibilities for parallel execution in R. In this work, we have chosen the foreach package [27]. Foreach is an iteration construct that transparently allows for each loop in the iteration to be executed in different cores or machines. Parallel execution can be achieved via doMC (using parallel/multicore on single workstations), doSNOW (using snow, which allows creating clusters made of several machines), or doMPI (using Rmpi and MPI or Message Passing Interface). In this work, three PCs with i7-3770 3.4 GHz and 10-Gb RAM have been used. Packages foreach and doMC [28] have been used for parallelizing work using the four real cores of the three i7-3770 CPUs.

4. PART I: STUDY OF MACHINE LEARNING TECHNIQUES FOR SOLAR ENERGY FORECASTING

In this part, SMO and GBR have been tested to approximate the solar energy production in the 98 mesonet stations given in the problem. First, for each mesonet station, models have been built using the information provided by the nearest grid nodes (from 1 to 16, incrementally), and then, an attribute selection algorithm is applied. Before running the models, some preliminary studies have been carried out in order to decide aspects related with the information provided by GEFS and to decide some important parameters of the machine learning methods.

4.1. Preliminary studies and parameter adjustment

As it has been mentioned in Section 2, data provided by GEFS include 11 ensemble output forecasting models. Using the 11 ensemble members as input variables to machine learning algorithms would imply to build up 11 regressors for each solar station. On the other hand, it is not obvious which ensemble member should be chosen. In this work, three different approaches to combine the 11 ensemble members have been considered: compute the mean of the 11 ensembles, compute the median, and compute the mode. The three approaches have been run using the information provided by the 5 nearest grid points and the average of mean absolute error (MAE) for the 98 mesonet stations are 1,940,816, 1,955,128, and 1,979,554, respectively. Therefore, we have decided to use the mean of the 11 ensemble models to summarize the information provided by all the ensembles.

On the other hand, the accuracy of SMO and GBR models depends on their parameters. To establish the optimal parameter values for each solar station and for each possible number of grid nodes would involve a very heavy computation. Then, the parameters of models have been selected using only the first of the 98 mesonet (ACME station). A 2-year validation dataset has been used to compare the different parameter combinations. An exhaustive grid search has been run to locate the optimal parameters (the cost parameter C and G for SMO and number of trees, shrinkage, and tree depth for GBR). For SMO with linear kernel (linear-SMO), parameter C has been tested for the following values: C $\in$ $(0.01, 0.03, 0.06, 0.12, 0.25, 0.50, 1.00, 2.00, 4.00)$. For SMO with RBF kernel (RBF-SMO), parameter C follows the same range, and $G \in (0.005, 0.010, 0.020, 0.040, 0.080, 0.160, 0.320, 0.640, 1.280)$. Experiments established that for linear-SMO, the best parameter is $C = 0.03$. For RBF-SMO, the best parameters are $C = 1$ and $G = 0.01$. GBR parameters have been tested for the following values: Trees $\in (1000, 3000, 5000, 7000)$, Shrink $\in (0.001, 0.005, 0.01, 0.1)$, and Depth $\in (6, 8, 10, 12, 14, 16)$. For GBR models, the optimal parameters are: number of trees $= 5000$, shrinkage $= 0.01$, and tree depth $= 10$. Those parameters have been used for all the experiments in this section.

4.2. Prediction accuracy with respect to the number of GEFS grid nodes

Figure 2 displays the evolution of MAE as the number of GEFS grid points is increased from 1 to 16 for linear-SMO, RBF-SMO, and GBR. Averaged train and test MAEs for 98 solar sites are shown on the left and right sides of Figure 2, respectively.

With respect to test MAE, it can be seen that the two non-linear models GBR and RBF-SMO perform significantly better than the linear one (linear-SMO). In all cases, it is observed that MAE tends to improve as the number of grid points increases. Both RBF-SMO and GBR obtain similar results when the number of grid points is large (8 or more). But GBR performs better when only a few grid points are used (from 1 to 4) and does not suffer from the slight overfitting observed for SMO for more than 10/11 GEFS grid points.

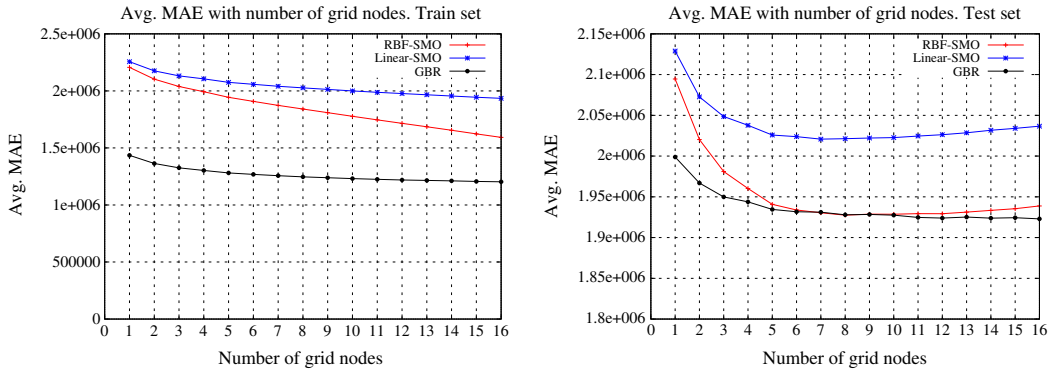

Figure 2. Average mean absolute error (MAE) for different number of grid nodes, using sequential minimal optimization (SMO) with linear kernel (linear-SMO), SMO with radial basis function kernel (RBF-SMO), and gradient boosted regression (GBR). Training and testing set.

The main conclusions from this study are that non-linear models perform much better than the linear one and that, interestingly, the best results are obtained using more than the closest four or five grid nodes (i.e., the grid nodes surrounding the station): the minimum error is obtained from eight grid points for RBF-SMO and from 16 points for GBR (although the gain obtained by GBR from 8 to 16 points is very small: a 0.26% decrease).

4.3. Study of feature selection methods

Here, the three feature selection methods have been applied to all the features present in 16 grid points ($16 * 75 = 1200$ features). The 1200 attributes are ranked, and both RBF-SMO and GBR algorithms are trained and tested using the first 400, 500, 600, 800, 900, and 1000 attributes, respectively. Figures 3 and 4 display the average MAE obtained for training and testing for the 98 stations using the different numbers of attributes.

Results show that, surprisingly, although the original number of attributes is very large, the different attribute selection methods do not improve prediction error in general. Therefore, in this domain, all 1200 attributes seem relevant to some degree. However, results also show that the number of attributes can be greatly reduced without losing a significant accuracy. In the case of RBF-SMO, LIA allows us to reduce the number of attributes from 1200 to 600 and obtain the same error (1,938,241 with 600 features vs. 1,938,855 with all features). In this case, ReliefF and linear correlation obtain higher errors for the same number of (600) attributes (1,965,442 and 1,997,274, respectively). When GBR is used as regressor, the number of features cannot be reduced to the same extent, but with 800

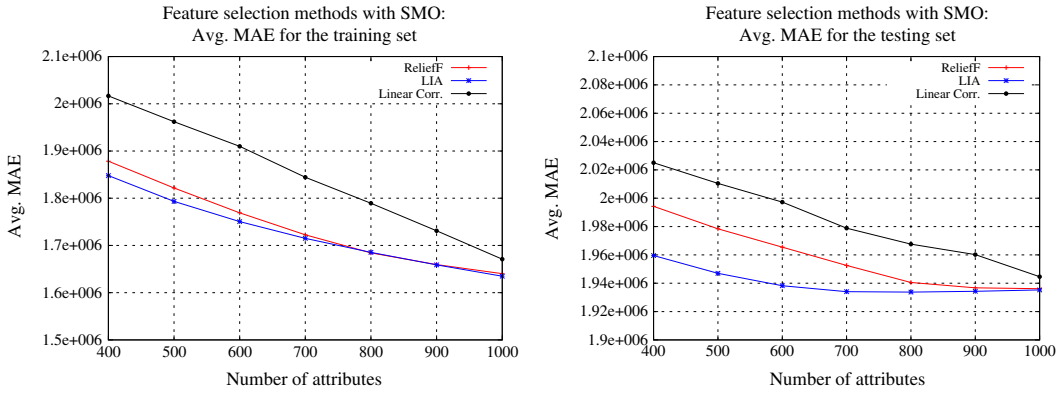

Figure 3. Average mean absolute error (MAE) for different number of attributes, using sequential minimal optimization (SMO) with radial basis function kernel (RBF-SMO). Training and testing set. LIA, local information analysis.

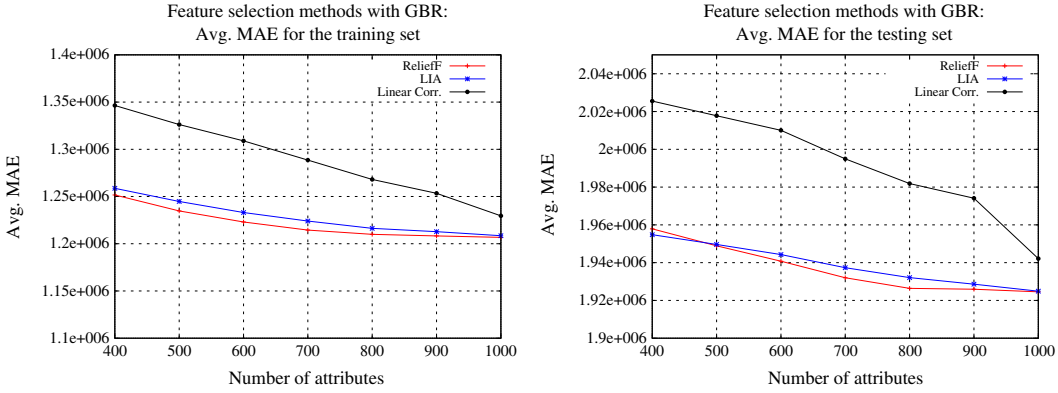

Figure 4. Average mean absolute error (MAE) for different number of attributes, using gradient boosted regression (GBR). Training and testing set. LIA, local information analysis.

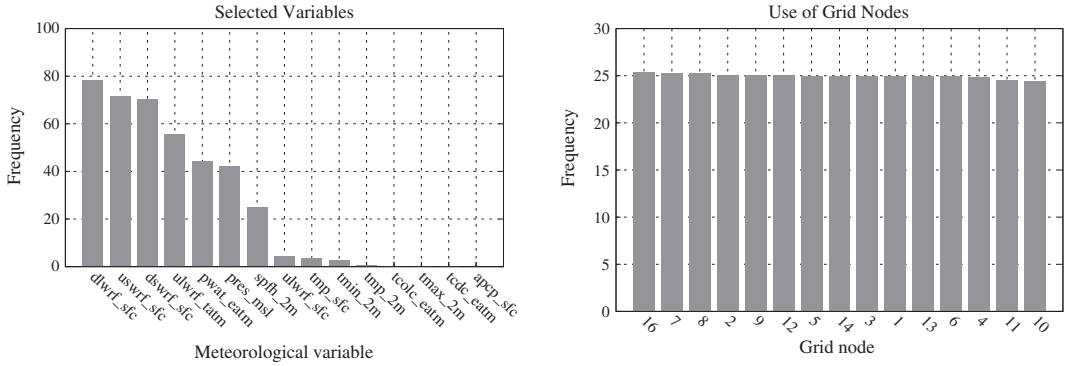

Figure 5. (a) Frequency of each meteorological variable in the 400 most relevant features; and (b) frequency of each Global Ensemble Forecast System grid node (1 = closest, 16 = farthest) in the 400 most relevant features.

features, ReliefF and LIA algorithms are able to obtain quite similar errors compared with the full set of features (1,926,369 and 1,932,069, respectively, versus 1,922,594 for the 1200 features). In all cases, linear correlation is not competitive with the other methods.

Finally, we will take advantage of the attribute ranking performed by the attribute selection methods in order to know which are the most relevant meteorological variables and the most relevant grid points. For this purpose, we have selected the 400 most relevant features, and we have used the attributes selected by LIA algorithm because it provides the best results for that number of features. Figure 5 displays bar graphs of the variable names and the grid points used, from 1 (the closest) to 16, respectively. Every feature is composed of a meteorological variable name, a time of day (from 1 to 5), and a grid node number (from 1 to 16). Then, the bar graph of Figure 5(a) displays the frequency of each meteorological variable name in the 400 attributes and Figure 5(b) displays the frequency of each grid node in the 400 attributes. Figure 5(a) shows that some of the variables appear more frequently and therefore are more important for forecasting. Some of these important variables are downward long-wave radiative flux average at the surface, upward short-wave radiation at the surface, downward short-wave radiative flux average at the surface, and upward long-wave radiation at the top of the atmosphere. On the other hand, the flatness of Figure 5(b) shows that all grid nodes have a similar frequency in the 400 features, and therefore, no preference is shown for closer versus farther away grid nodes, all seem equally important.

5. PART II: AGGREGATE MODELS FOR SOLAR ENERGY FORECASTING

The aims of this section were to describe how aggregate models are constructed, to compare them with individual models (obtained in the previous section), and to experimentally test them with the purpose of forecasting at new locations.

5.1. Aggregate models

In the previous section, individual models were constructed for each station, and it was concluded that all the 16 grid nodes around each station supply relevant information. These 16 grid nodes include the four grid nodes surrounding the station (forming the inner square) and the 12 nodes that, in turn, surround the former four nodes (i.e., the outer square). Given that all the stations within the inner square share the same input data (i.e., the meteorological variables of the 16 grid nodes), we propose to build a single model, called aggregate model, for each square in the grid. The data sets for each aggregate model are constructed by joining the data associated with each station within the square. Therefore, the number of instances for training is 4383 days times the number of stations within each square. The output variable in the training set corresponds to the radiation of each station in the square. The input attributes are the five-time instants of each of the 15 meteorological

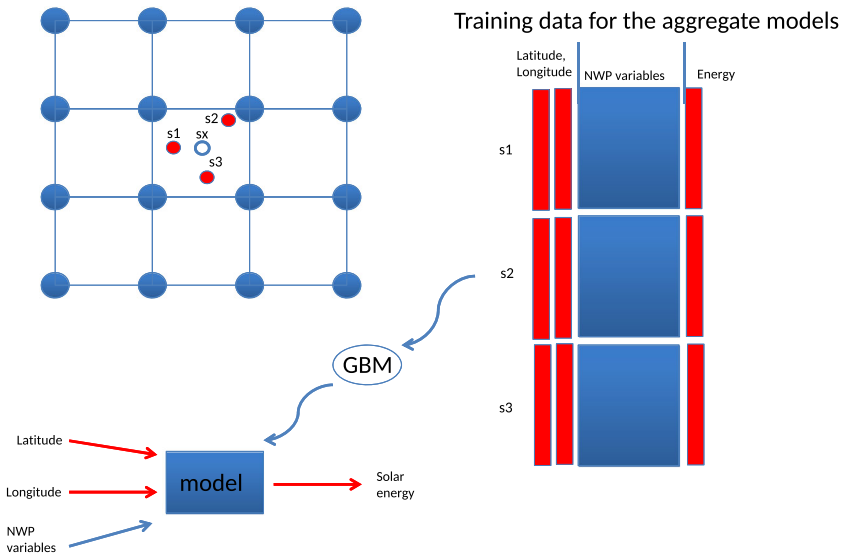

Figure 6. Building of aggregate models. NWP, numerical weather prediction.

variables for each one of the 16 nodes surrounding each square ($5 \times 15 \times 16 = 1200$ attributes), like in the individual models. But in addition, the latitude and longitude of each station are added to the set of attributes, so that information about station locations is given to the model. Figure 6 illustrates the construction of the aggregate model for a square with three stations (s_1, s_2, and s_3) that could be used to predict in location s_x. Aggregate model in a square can be used to predict the solar radiation in all the stations within the square, and thus, the number of models will be reduced, from 98 (stations) to 21 (squares). But more importantly, aggregate models will be able to predict solar radiation for any location (any latitude and longitude within the square), not only for those stations in the training set.

Aggregate models will be evaluated in two ways. First, they will be used to predict solar energy in each station, just as the individual models, with the purpose of checking whether they have a reasonable performance. Predictions with aggregate models will be made for every solar station within its square by supplying its latitude and longitude to the model and compared with the predictions of the respective individual model. Years 1997 to 2005 (from all the stations within a square) are used for training and 2006 to 2007 for testing, similarly to the individual models. For instance, the square whose bottom-left coordinates are (34,−99) contains three solar stations (stations 1, 4, and 57). Years 1997 to 2005 from the three stations will be used to construct the aggregate model, which will be tested on data for the three stations, from 2006 to 2007.

Second, aggregate models will be evaluated for forecasting at new locations. In this case, aggregate models should be learned from some of the stations within the square, but tested on other stations within the square, different to the ones used for learning. Thus, the methodology for evaluating the performance of interpolation models has been similar to cross-validation (and more specifically, leave-one-out) but applied to stations instead of data instances. That is, given a square in the grid, the aggregate model is trained with all stations within the square but one, the latter being used for testing. This is carried out as many times as stations within the square; each time, one station is being used for testing. As before, years from 1997 to 2005 are used for training the aggregate models and from 2006 to 2007 for testing them. If there are N stations within the square, the final result is computed by averaging the N validation values. For instance, for square (34,−99) with stations 1, 4, and 57, three aggregate models will be constructed. The first one will use training data from stations 1 and 4, from years 1997 to 2005, and it will be tested on data from station 57, from years 2006 to 2007. The second one will use stations 1 and 57 for training and 4 for testing, and so on.

Aggregate models have been built using GBR (from the study in Section 4, it is the best algorithm regression). Similarly to individual models, GBR parameters need to be adjusted to

the new aggregate training data sets. As in Part I, parameters are number of trees, regularization parameter (shrinkage) and depth of each tree (Trees, Shrink, and Depth). Therefore, before building the final models, a study has been made of the parameters to use, using an exhaustive grid search. Seventy percent of the training data have been used for building the model and 30% for evaluating it. As in the first part, all combinations of the parameters have been tested for the following values: Trees $\in (1000, 3000, 5000, 7000)$, Shrink $\in (0.001, 0.005, 0.01, 0.1)$, and Depth $\in (6, 8, 10, 12, 14, 16)$. The two best configurations had Shrink $= 0.005$ and Depth $= 14$, with 7000 and 5000 trees, respectively. Because the difference between them was not large, the simplest configuration was chosen (5000 trees). This study was conducted with the square whose bottom-left point corresponds to latitude 36 and longitude -103 (in Figure 1, point with coordinates (4,6) of the grid), and the same configuration was used for the rest of squares. As in Part I, GBR is set to optimize the MAE.

5.2. Experimental results

Table I displays for each of the squares, the testing MAE obtained for the individual models (see column 'Individual models'), the testing MAE of the aggregate models used for prediction (see column 'Aggregate models'), and the testing MAE of the aggregate models used for interpolation (see column 'Aggregate interpolation'). The number of stations within each square is also included. The bottom row displays the overall average MAE obtained by each approach (bearing in mind that it is a weighted average, because different models contain a different number of stations). Models for squares 10, 14, and 21 have not been considered, because they only include one station (therefore, the individual and aggregate models would use the same data).

With respect to aggregate models for prediction, it can be observed that the average error is very similar to the individual models (only 0.34% worse). Therefore, the performance of aggregate models for prediction is appropriate, and it is reasonable their further use for forecasting at new locations. With respect to aggregate models for interpolation, it can be observed that the average error is higher than the one obtained for prediction (1,961,738 vs. 1,929,169). This is to be expected,

Table I. Individual models versus aggregate models.

Model number	Number of stations	Individual models	Aggregate models	Aggregate interpolation
1	2	1,601,105.22	1,598,086.75	1,627,001.99
2	2	1,949,703.18	1,963,578.90	2,007,125.00
3	2	1,848,804.12	1,829,587.25	1,834,192.56
4	5	1,920,817.32	1,926,013.42	1,966,784.78
5	5	1,803,397.20	1,799,948.66	1,822,806.88
6	7	1,810,209.57	1,813,106.64	1,835,861.82
7	3	1,844,588.10	1,847,715.74	1,863,821.08
8	6	1,770,555.56	1,772,931.67	1,786,517.55
9	3	1,878,825.75	1,852,969.14	1,869,272.21
10	1	—	—	—
11	7	1,885,825.48	1,892,303.56	1,924,896.06
12	5	1,822,892.06	1,830,952.20	1,869,984.70
13	6	1,825,561.94	1,847,732.68	1,882,323.54
14	1	—	—	—
15	6	2,066,941.63	2,078,522.80	2,097,825.57
16	5	1,917,603.84	1,912,938.86	1,936,991.07
17	7	1,892,978.65	1,931,564.55	1,946,798.10
18	7	2,077,039.15	2,074,416.33	2,113,034.47
19	5	2,192,207.18	2,198,952.64	2,225,971.53
20	4	2,008,902.16	2,013,784.37	2,042,272.60
21	1	—	—	—
22	2	2,204,235.19	2,219,038.80	2,506,957.16
23	3	2,039,895.71	2,052,441.62	2,068,843.74
24	3	2,048,404.49	2,072,367.68	2,127,614.00
		1,922,511.51	**1,929,169.40**	**1,9617,38.19**

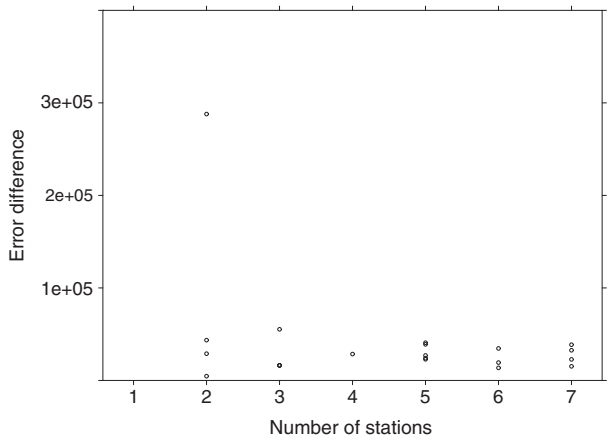

Figure 7. Error differences between aggregate models for interpolation and for prediction versus number of stations in the square.

because interpolation does not use as training data information from the station that is being used to measure the interpolation capability. In any case, the error obtained when interpolating across the square is acceptable and does not differ in excess from the error obtained by individual models (39,227 J $\times$ m^{-2}) and by the aggregate models (32,569 J $\times$ m^{-2}), approximately 2% in both cases. The largest difference is observed in model 22, about 300,000 J$\times$m^{-2}. This model corresponds to the model built for the square whose bottom-left corner corresponds to latitude 34 and longitude -95. In Figure 1, the point corresponds to coordinates (12,4) of the grid, and it can be seen that the two stations belonging to that square are far apart. Figure 7 plots the difference between the aggregate models for interpolation and for prediction versus the number of solar stations within the square. As it is observed, excluding the largest difference mentioned before for model 22 (with two stations), the rest of the differences are very similar and do not depend on the number of stations within the square.

6. CONCLUSIONS

In this work, prediction of solar energy from meteorological variables provided by NWP models is addressed. Data used in this study come from the National Oceanic and Atmospheric Administration/Earth System Research Laboratory GEFS where the problem consists on forecasting solar energy at 98 Oklahoma Mesonet solar sites using 15 NWP variables on a 16×9 spatial grid. In this context, the aim of this article is twofold. First, to determine the influence of the number of grid nodes on forecasting accuracy of three different regression methods (linear SVM, RBF-SVM, and GBR). Also, given the large number of features in this domain, three different attribute selection methods have been tested (linear correlation, ReliefF, and LIA) with the purpose to extract the most relevant information (number of nodes in the grid and meteorological variables) to predict the solar energy in the 98 stations.

The second aim of this paper is to construct models not only for the solar stations used for training the models but also for any new solar site that could be deployed on a new location in the future. The models of the first part can be used to predict the solar energy in one of the stations used for training the model, but not for new different locations. With this aim, in this paper, aggregate models are built up for square regions in the grid, using the data of all the stations within the square and including as input to the models the latitude and longitude of the stations. Square regions have been used because all stations within the same square share all the meteorological variables of the 16 surrounding nodes. Those models can be used to predict the solar energy not only for the stations in the square but also for any new latitude and longitude.

With regard to the first issue, experimental results show that the non-linear methods obtain lower errors than the linear one. The non-linear models, GBR and RBF-SMO, perform similarly, although

RBF-SMO shows some slight overfitting when the number of grid points is large. Also, in the case of the best performing method (GBR), forecasting accuracy tends to improve as the number of GEFS grid nodes used as input increases, even beyond the four or five closest nodes. In the case of the GBR method, the best performance is obtained with information from at least 11 or 12 nodes, not just the four nodes around the station. Beyond 12 nodes, improvements become very small, and the trend suggests that using more than 16 nodes would not increase accuracy significantly. Contrary to what was expected, feature selection was not able to improve solar energy prediction, although with RBF-SMO, LIA can obtain similar predictions with half the attributes. However, for the best method, GBR, the best performance is obtained with all the attributes of the 16 grid nodes (1200). For this method, ReliefF algorithm performs slightly better than LIA, but the best performance is obtained with the set of all attributes. Hence, from this study, it is concluded that the best method is GBR, and the information provided by the 16 nodes around the station is relevant to obtain the best performance.

With respect to the second goal, the results obtained with aggregate models to predict solar energy in each of the 98 stations are very similar to the results obtained with individual models, showing the validity of the aggregate models. At the same time, given that latitude and longitude are inputs to the aggregate models, this approach allows us to use the model to make forecasts for new locations. To validate the latter, a similar methodology to leave-one-out has been used, where each station within the square will be used, in turn, as testing location and the rest of the stations within the square will be used to train the aggregate model. As the historical data of the testing solar station are not used to build the interpolation aggregate model, errors obtained are higher than those obtained with the aggregate model for prediction, as expected, although the increase is relatively small. Therefore, replacement of individual models by models that cover certain geographic areas is an interesting alternative to predict not only stations already known but also new locations.

ACKNOWLEDGEMENT

The authors acknowledge financial support granted by the Spanish Ministry of Science under contract ENE2014-56126-C2-2-R (AOPRIN-SOL).

REFERENCES

1. Diagne M, David M, Lauret P, Boland J, Schmutz N. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews* 2013; **27**:65–76.
2. Mellit A, Pavan AM. A 24-h forecast of solar irradiance using artificial neural network: application for performance prediction of a grid-connected {PV} plant at Trieste, Italy. *Solar Energy* 2010; **84**(5):807–821.
3. Sharma N, Sharma P, Irwin D, Shenoy P. Predicting solar generation from weather forecasts using machine learning. *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on,* IEEE, 2011; 528–533.
4. Chen JL, Liu HB, Wu W, Xie DT. Estimation of monthly solar radiation from measured temperatures using support vector machines—a case study. *Renewable Energy* 2011; **36**(1):413–420.
5. Alaíz CM, Torres A, Dorronsoro JR. Sparse linear wind farm energy forecast. *Icann (2)*, 2012; 557–564.
6. Monteiro C, Bessa R, Miranda V, Botterud A, Wang J, Conzelmann G. Wind power forecasting: state-of-the-art 2009. *Technical Report*, Argonne National Laboratory (ANL), 2009.
7. Wolff B, Lorenz E, Kramer O. Statistical learning for short-term photovoltaic power predictions. *Dare: Data Analytics for Renewable Energy Integration. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Prague, 2013.
8. Gala Y, Fernandez A, Dorronsoro JR. Machine learning prediction of global photovoltaic energy in Spain. *International Conference on Renewable Energies and Power Quality*, 2014.
9. Aler R, Martín R, Valls JM, Galván IM. A study of machine learning techniques for daily solar energy forecasting using numerical weather models. In *Intelligent Distributed Computing VIII*. Springer, 2015; 269–278.
10. Cortes C, Vapnik V. Support-vector networks. *Machine Learning* 1995; **20**(3):273–297.
11. Friedman JH. Greedy function approximation: a gradient boosting machine. *Annals of Statistics* 2001:1189–1232.
12. Friedman JH. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 2002; **38**(4):367–378.
13. Liu H, Sun J, Liu L, Zhang H. Feature selection with dynamic mutual information. *Pattern Recognition* 2009; **42**(7):1330–1339.
14. Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. *Bioinformatics* 2007; **23**(19):2507–2517.

15. Kononenko I. Estimating attributes: analysis and extensions of relief. *Machine Learning: Ecml-94*, Springer, 1994; 171–182.
16. Keskin M, Dogru AO, Balcik FB, Goksel C, Ulugtekin N, Sozen S. Comparing spatial interpolation methods for mapping meteorological data in Turkey. In *Energy Systems and Management*. Springer, 2015; 33–42.
17. Webster Richard, Oliver Margaret A. *Geostatistics for Environmental Scientists*. John Wiley & Sons. Page 149, 2007.
18. González-Longatt F, Medina H, González JS. Spatial interpolation and orographic correction to estimate wind energy resource in Venezuela. *Renewable and Sustainable Energy Reviews* 2015; **48**:1–16.
19. Alsamamra H, Ruiz-Arias JA, Pozo-Vázquez D, Tovar-Pescador J. A comparative study of ordinary and residual kriging techniques for mapping global solar radiation over southern Spain. *Agricultural and Forest meteorology* 2009; **149**(8):1343–1357.
20. Mubiru J, Banda EJKB. Monthly average daily global solar irradiation maps for Uganda: a location in the equatorial region. *Renewable Energy* 2012; **41**:412–415.
21. Burges CJC. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 1998; **2**(2):121–167.
22. Vapnik VN. *Statistical Learning Theory (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. John Wiley & Sons, A Wiley-Interscience Publication: New York, 1998.
23. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The Weka data mining software: an update. *ACM SIGKDD Explorations Newsletter* 2009; **11**(1):10–18.
24. Schonlau M. Boosted regression (boosting): an introductory tutorial and a Stata plugin. *Stata Journal* 2005; **5**(3):330.
25. Greg R, with contributions from others. *gbm: Generalized Boosted Regression Models*, 2013. R package version 2.1.
26. R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2014.
27. Analytics Revolution, Weston Steve. *Foreach: Foreach Looping Construct for R*, 2014. R package version 1.4.2.
28. Analytics Revolution. *doMC: Foreach Parallel Adaptor for the Multicore Package*, 2014. R package version 1.3.3.