Computers and Chemical Engineering 101 (2017) 23–30

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Comparative analysis of data mining and response surface methodology predictive models for enzymatic hydrolysis of pretreated olive tree biomass

Francisco Charte[a], Inmaculada Romero[b], María D. Pérez-Godoy[a], Antonio J. Rivera[a], Eulogio Castro[b,*]

[a] Dpt. Computer Science, Universidad de Jaén, Jaén, Spain
[b] Dpt. Chemical, Environmental and Materials Engineering, Universidad de Jaén, Jaén, Spain

ARTICLE INFO

Article history:
Received 24 November 2016
Received in revised form 11 January 2017
Accepted 4 February 2017
Available online 17 February 2017

Keywords:
Predictive models
Data mining
Enzymatic hydrolisis
Olive tree biomass

ABSTRACT

The production of biofuels is a process that requires the adjustment of multiple parameters. Performing experiments in which these parameters are changed and the outputs are analyzed is imperative, but the cost of these tests limits their number. For this reason, it is important to design models that can predict the different outputs with changing inputs, reducing the number of actual experiments to be completed. Response Surface Methodology (RSM) is one of the most common methods for this task, but machine learning algorithms represent an interesting alternative. In the present study the predictive performance of multiple models built from the same problem data are compared: the production of bioethanol from lignocellulosic materials. Four machine learning algorithms, including two neural networks, a support vector machine and a fuzzy system, together with the RSM method, are analyzed. Results show that Reg-CO^{2}RBFN, the method designed by the authors, improves the results of all other alternatives.

1. Introduction

The production of bioethanol from lignocellulosic materials (LCM) such as agricultural, agroindustrial or other biomass residues has been revealed as a promising alternative for partially substituting fossil fuels (Behera et al., 2014).

Pretreatment, enzymatic hydrolysis and fermentation are the three basic steps of the conversion scheme, which in summary consists in the transformation of simple sugars contained in the LCM into ethanol. The main objective of the pretreatment is to break down the lignocellulosic structure of the raw materials, resulting in an improved access of enzymes to the sugar bonds. Next, the enzymatic hydrolysis itself will produce simple sugars (monomers, like glucose, xylose and others) from their polymers (cellulose chains and hemicelluloses). Finally, the action of a fermenting microorganism will transform the monomers into ethanol, which will be later separated from the fermentation broth. The design and validation of accurate mathematical models, able to consider the main operational steps of the conversion process, is a key factor for taking full advantage of the use of renewable energy sources with that purpose.

Several factors can affect the performance of the pretreatment, including temperature, processing time, and the use of salts (Ravindran and Jaiswal, 2016). As a general rule, severe pretreatment conditions will result in an easy to hydrolyze pretreated solid, but will also lead to higher material loss. A compromise selection of operational conditions considering energetic requirements of pretreatment, recovery of pretreated material, and ease of hydrolysis is necessary. The enzymatic hydrolysis of the resulting pretreated materials can be used as a guide for condition selection.

To take account of the multiple variables or factors involved in the pretreatment of LCM while keeping the experimental work charge in a reasonable point, experimental designs are usually applied. In experimental designs, all the factors are simultaneously changed from one experiment to the other, in opposition to the one-factor-at-a-time strategy, which can lead to a much higher number of experiments (Bezerra et al., 2008). Once the experiments are performed, a mathematical model is developed to relate the experimental variables (or factors) with the dependent variable(s) (or response). The response surface methodology (RSM), which is the usual method applied in this field, consists in analyzing such a relationship through a second degree polynomial.

* Corresponding author.
E-mail addresses: fcharte@ujaen.es (F. Charte), iromero@ujaen.es (I. Romero), lperez@ujaen.es (M.D. Pérez-Godoy), arivera@ujaen.es (A.J. Rivera), ecastro@ujaen.es (E. Castro).

http://dx.doi.org/10.1016/j.compchemeng.2017.02.008
0098-1354/

Data Mining (DM) can be considered as a central stage within the more general knowledge discovery process (Maimon and Rokach, 2010). The objective of DM is to determine models that include the relations among data and allow a better understanding, prediction or generalization of these data. One of the most important DM tasks is regression, i.e. obtaining models with n independent variables and a continuous dependent variable. Successful applications of different DM methods for regression tasks can be found in the bibliography. For instance, Artificial Neural Networks (ANNs) are used in Cao et al. (2016) to model a desalination process or in Ochoa-Estopier et al. (2013) to optimize a distillation system. In Zhao et al. (2016) a Support Vector Machine (SVM) is used to predict the viscosity of ionic liquids. Fuzzy systems are used in Wang et al. (2016) to model the component concentrations in optical gas systems.

The objective of this work is to compare through RSM and DM regression methods the results obtained from the enzymatic hydrolysis of pretreated olive tree biomass (OTB), an abundant, low cost and lacking of industrial application agricultural residue of great importance in Mediterranean countries.

Among the main contributions in the present work the following ones must be highlighted: First, the use of different DM methods, not only ANNs but also SVMs and Fuzzy Systems. One of the ANN methods (Reg-CO^{2}RBFN) was developed and adapted by the authors for regression tasks. Second, to ensure the reliability of the results data are partitioned following the cross validation approach, performing 20 repetitions of the experiments on as many different data partitions. Third, to carry out a formal statistical analysis, a two stage methodology was applied: firstly, the Friedman test (Sheskin, 2006) is used to detect statistical differences among results. If so, the second step consists in applying the Holm test (Holm, 1979), which is a post-hoc procedure aimed to concretize the significance of these differences.

Some of the DM methods tested in the experimental section, specifically ANNs such as Radial Basis Function Networks (RBFNs) or Multilayer Percentrons (Rojas, 1996) (MLPs), have already achieved successful results in prior experiments. The RBFNs are designed with a cooperative-competitive methodology developed by the authors (Reg-CO^{2}RBFN). Additionally, SVMs as NU-SVR (Fan et al., 2005) and Fuzzy Systems developed with Genetic Programming as GFS-GAP-Sys (Sánchez and Couso, 2000) were tested. As summary, Reg-CO^{2}RBFN achieved the best results outperforming with significant differences the remaining methods, including RSM which is the one most usually applied in the field.

2. Materials and methods

2.1. Pretreatment of materials

Fifteen pretreatment experiments with OTB were performed according to a Box-Behnken experimental design fully described in López-Linares et al. (2013). Briefly, operational conditions included the duration (Time, min) and temperature (Temp, C) of the pretreatment, and the concentration of $FeCl_3$ (C, M). The respective ranges of these factors were 0–30 min, 120–180 C, and 0.050–0.275 M. Once the pretreatment experiments were done, liquids and pretreated solids were separated by filtration, and the content of glucose and hemicellulosic sugars in both fractions was determined. Furthermore, the pretreated solids were further submitted to enzymatic hydrolysis under standard conditions, and the enzymatic hydrolysis yield for each experiment (grams of glucose in the hydrolysate per gram of glucose in the pretreated material or in the original material) was evaluated. Specifically the output variables obtained are: recovery (%) of total solids (SR), glucose in pretreated solids (GRS), glucose in prehydroLisates (GRL), hemicellulosic sugars in pretreated solids (HSRS), hemicellulosic sugars in prehydroLisates (HSRL), enzymatic hydrolisis yields in raw material (YEH) and enzymatic hydrolisis yields in pretreated material (YWIS). The values for these variables in the conducted experiments can be seen in Table 1.

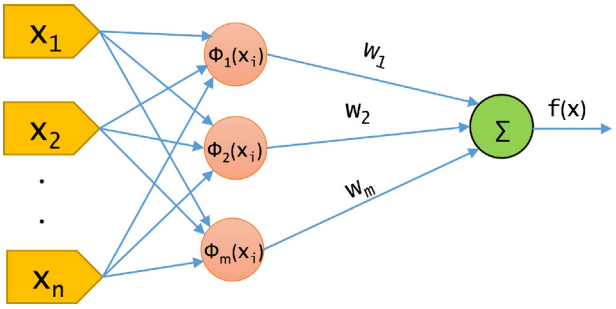


Fig. 1. RBFN topology for regression problems.

2.2. The RSM model

Once factors (temperature, time and salt concentration) and responses (solid recovery, glucose recovery in pretreated solids and liquids, hemicellulosic sugar recovery in solids and liquids, and enzymatic hydrolysis yields) were defined and the experiments were performed, a polynomial equation relating each response with the factors was obtained, as follows:

$$Y = a_0 + a_1T + a_2C + a_3t + a_4TC + a_5Tt + a_6Ct + a_7T^2 + a_8C^2 + a_9t^2 \quad (1)$$

where Y stands for each of the responses and the a-terms are to be determined so that the experimental results are best adjusted according to different statistical parameters such as the correlation coefficient R^2. The graphical representation of Eq. (1), when one of the factors is set in a particular value, corresponds to a surface from which the maximum or minimum can be obtained.

2.3. Machine learning methods

Four learning machine methods have been used to process the collected data. The goal is to compare the performance of the usual RSM model against other potential solutions. The selected methods are: Reg-CO^{2}RBFN, our developed method, MLP-BR (Rojas, 1996), NU-SVR (Fan et al., 2005) and GFS-GAP-Sys[1] (Sánchez and Couso, 2000). In this subsection the foundations of these machine learning algorithms are briefly introduced.

2.3.1. The Reg-CO^{2}RBFN method

Together with MLPs, RBFNs (Broomhead and Lowe, 1988) are one of the most well known artificial neural network paradigm. Among their characteristics highlight a simple topological structure and universal approximation ability (Park and Sandberg, 1993).

The topology of an RBFN is composed by three feed-forward connected layers: an input layer with n nodes (as many as inputs there are), a hidden layer with m neurons or RBFs, and an output layer with one node for regression problems (see Fig. 1).

The activation function of the neurons or RBFs in the hidden layer is a radially-symmetric basis function, $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}$. The shape of this function usually is the Gaussian one (Equation (2))

$$\phi_i(\vec{x}) = e^{-(\|\vec{x}-\vec{c}_i\|/d_i)^2} \quad (2)$$

[1] These are the methods' names in the KEEL software package (Alcalá-Fdez et al., 2011), whose implementation has been used in this study.

Table 1
Values from the OTB pretreatment experiments.

	Input variables			Output variables						
n	Time	Temp	FeCl3	GRL	GRS	HSRS	HSRL	SR	YEH	YWIS
1	30	120	0.125	32.48	51.14	22.68	60.63	60.16	10.14	19.71
2	0	140	0.050	29.67	61.92	55.22	28.46	71.36	12.23	19.92
3	15	160	0.050	35.73	56.54	15.20	73.34	56.28	19.57	34.61
4	0	140	0.200	35.01	59.50	17.52	67.98	57.74	11.66	19.43
5	30	140	0.200	35.31	49.25	0.33	82.20	49.21	19.67	39.94
6	15	140	0.125	33.01	63.10	12.38	71.69	54.46	14.62	23.17
7	15	140	0.125	32.66	55.22	13.02	73.49	55.05	15.1	27.34
8	30	160	0.125	33.14	50.30	0.00	62.70	48.37	28.07	55.80
9	15	120	0.050	27.45	62.87	66.65	28.46	73.83	11.85	18.85
10	30	140	0.050	31.00	55.92	37.56	44.96	63.07	11.08	19.81
11	15	120	0.200	32.31	59.86	20.34	72.15	57.38	8.76	14.63
12	0	160	0.125	37.35	55.81	5.60	81.02	53.96	18.80	33.69
13	15	140	0.125	33.55	54.86	13.07	70.14	55.48	14.10	25.70
14	15	160	0.200	32.01	48.32	0.00	50.35	46.51	36.50	75.54
15	0	120	0.125	27.39	62.87	64.00	31.17	70.83	12.62	20.07
16	0	180	0.200	21.88	27.46	0.00	15.38	42.26	25.58	93.16
17	0	160	0.275	28.16	50.09	0.00	28.84	44.27	28.73	66.87
18	30	180	0.200	28.15	21.45	1.85	7.13	40.66	20.17	93.55
19	30	160	0.275	41.24	43.80	0.00	40.57	45.16	38.85	88.71
20	30	180	0.275	18.45	11.81	0.00	2.44	41.34	11.36	96.15

where $\vec{c}_i \in \mathbb{R}^n$ is the center of basis function ϕ_i, $d_i \in \mathbb{R}$ is the radius or width, and $\|\|$ is typically the Euclidean norm on $\mathbb{R}^n$.

The output neuron calculates the sum of all RBF outputs pondered by the corresponding weights, w_i, see Equation (3)

$$f(\vec{x}) = \sum_{i=1}^{m} w_i \phi_i(\vec{x}) \tag{3}$$

The authors have developed an hybrid evolutionary cooperative-competitive methodology for the design of RBFNs, CO^{2}RBFN (Pérez-Godoy et al., 2010) for classification tasks. Now, in this paper, a new adapted version for regression problems, named Reg-CO^{2}RBFN, is proposed.

In Reg-CO^{2}RBFN, an individual of the population represents an RBF and the entire population the solution to the problem. The quality or credit assignment of an individual is measured with three parameters: the RBF weight to the network output, a_i, the error into the basis function radius, e_i, and the overlapping, o_i, with remaining RBFs. Specifically for each RBF, these parameters are defined as:

- The contribution, a_i, of the RBF ϕ_i, $i = 1 \ldots m$, is established to the weight of the RBF.
- The error measure, e_i, for each RBF ϕ_i, is obtained calculating the RMSE (Root Mean Square Error) (Equation (5)) error of the patterns inside its radius.
- The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i. This factor is expressed as:

$$o_i = \sum_{j=1}^{m} o_{ij} \qquad o_{ij} = \begin{cases} (1 - \|\phi_i - \phi_j\|/d_i) & if \|\phi_i - \phi_j\| < d_i \\ 0 & otherwise \end{cases} \tag{4}$$

where o_{ij} measures the overlapping of the RBF ϕ_i y ϕ_j $j = 1 \ldots m$.

As part of the evolutionary environment four operators can be applied to the individuals: removing, random mutation, biased mutation and no operation. These operators are applied in function of a Fuzzy Rule-Based System (FRBS), where the inputs are the three parameters defined for credit assignment and the outputs are the probability of applying the operators. The main steps of Reg-CO^{2}RBFN are shown in Algorithm 1 in pseudocode.

Algorithm 1. Reg-CO2RBFN pseudo-code.

```
RBFN ← initialState()                          ▷ Initialize RBFN
repeat                                         ▷ RBFN training process
    Evaluate(RBFs)
    RBFs ← applyOperators(RBFs)
    ▷ Substitute RBFs marked for removing
    RBFs ← RBFs - marked(RBFs)
    RBFs ← selectBest(RBFs)
until Stop condition is met
```

2.3.2. Symbolic regression through evolutionary techniques

GAs (*Genetic Algorithms*) and GP (*Genetic Programming*) (Affenzeller, 2009) are a family of optimization techniques based on evolutionary principles. In a GA each potential solution to the faced problem is described as an individual, whose genotype (a fixed-length string of binary or real values) encodes the solution parameters. The genotypes of the population are changed over time through cross-over and mutation operators, and a fitness evaluation selects the best adapted individuals through a tournament. After a certain number of generations the best solution, or a set of best solutions, is retrieved. GP is similar to GA in the way the solutions are evolved, but the representation of the individuals is completely different. A symbolic description of the solution, usually in the form of a tree, along with a certain set of restrictions that guarantee its validity, is used instead of a fixed-length string of symbols. This way, GP can be used to search for computer programs, arithmetic expressions or any other symbolic portrayal that solves the problem at glance.

GA-P (Howard and D'Angelo, Exper) is an evolutionary method able to face symbolic regression tasks. To do so, it combines GAs and GP techniques. The GA part is in charge of optimization, while the evolution of mathematical expressions relies on GP. GA-P's goal is to find data relationships which help to solve regression in a symbolic fashion. The GFS-GAP-Sys algorithm proposed in Sánchez and Couso (2000) is based on GA-P, but adapted to work with fuzzy data (Harris, 1989). The result is a fuzzy arithmetic-based model capable of extracting useful symbolic relations among inputs and outputs. GFS-GAP-Sys proved to be competitive against other methods, including artificial neural networks, in facing tasks such as the discovery of empirical laws from sets of data samples.

The most important configuration parameters for this algorithm are the population size, number of subpopulations (islands), the tournament size (number of individuals involved in the tournament), and the probability of a mutation being applied.

2.3.3. Multi-layer perceptron regression

MLP (*multi-layer perceptron*) is the best-known ANN model, and it is probably the most used one since the algorithm which allows to train it was introduced in Rumelhart et al. (1985). Unlike RBFNs, an MLP can have more than one layer of hidden units. Once a data pattern has been given as input, the activation function inside each unit is computed, and its output is forwarded to the next layer until reaching the output layer, where the predicted value is reported. During training the MLP, this predicted value is compared with the ground truth value, and the committed error is used to adjust the weights connecting the units in each layer. The error is minimized in each trip of values through the MLP layers. After presenting the input data several times to the algorithm, the MLP eventually reaches an equilibrium state. It has learned how to predict the correct output values from each set of inputs. MLPs are mostly applied to classification problems.

MLPs are universal estimators for continuous and bounded functions, so they can solve regression problems in a natural way. Its robustness and stability against the traditional statistical approach, i.e. linear regression, have been demonstrated (Gaudart et al., 2004). In classification tasks, a sigmoid activation function translates the continuous output into a categorical value, the class label. MLP-BR (Rojas, 1996), the algorithm used in our experiments, is essentially a multi-layer perceptron (MLP) designed to produce a continuous output as prediction, instead of a class label identifier.

The most important configuration parameters for this algorithm are the number of hidden nodes, the transfer function used by the neurons, the η and α values for the momentum term, and the λ value for the decay term.

2.3.4. Support vector machine regression

SVMs (*support vector machines*) (Boser et al., 30401) were originally applied to pattern recognition tasks. The input pattern features, lying in a n-dimensional space and which are non-linear separable, are projected into a larger space by means of a kernel function, achieving linear separability. The algorithm, through the resolution of a quadratic optimization problem, finds the maximum separation margin between pattern categories. Those patterns located in the frontier of two classes, helping to generate the separation boundary, are the support vectors. The traditional SVM method has a *C* parameter that is hard to optimize, since it is not bounded. The NU-SVM algorithm (Schölkopf et al., 2000) replaces the *C* parameter introducing a new one, called *NU*, which is bounded and has a straightforward interpretation, thus being easier to adjust.

SVMs can be seen as an universal tool (Vapnik et al., 1996) for solving any multidimensional function estimation, including regression approximation problems. The standard SVM method has a major drawback, the training process is quite expensive, since it implies solving a large quadratic optimization problem. NU-SVR (Fan et al., 2005) is a regression SVM based on the SMO (*Sequential Minimal Optimization*) (Platt, 1998) algorithm, reported to be orders of magnitude more efficient than the traditional learning algorithm.

The most important configuration parameters for this algorithm are the kernel type used to project the source space and some values related to that kernel function, such as the degree and coefficients of the polynomial.

3. Results and discussion

3.1. Experimental framework

The implementation of the data mining methods: FUZZY-GAP, MLP-BP and NU-SVM, has been obtained from KEEL (Alcalá-Fdez et al., 2011). The values of the parameters are set to the default ones. Since these three DM methods are not deterministic, three independent runs have been executed and average values have been gathered. The main parameters used for the algorithms are shown in Table 2.

Table 2
Parameter specification for the algorithms employed in the experimentation.

Algorithm	Parameter	Value
Reg-CO²RBFN	Generations of the main loop	100
	Number of RBFs	8
FUZZY-GAP	Population size	30
	Number of islands	2
	Steady	1
	Number of iterations	1000
	Tournament size	4
	Probability of mutation	0.01
MLP-BP	Hidden_nodes	8
	Transfer	Htan
	Eta	0.15
	Alpha	0.10
	Lambda	0.0
NU-SVM	KERNELtype	POLY
	C	100.0
	Eps	0.001
	Degree	1
	Gamma	0.01
	Coef0	0.0

For each output variable of the hydrolysis process, one model is determined for every regression or data mining method.

Two evaluation metrics have been computed to assess the methods performance. The first one is RMSE (*root mean square error*) (Eq. (5)), where n is the number of instances, f_t is the output of the model and y_t is the real output for the tth instance respectively.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(f_t - y_t)^2}{n}} \tag{5}$$

Another way to measure the quality of the calculated model from the training data is the coefficient of determination, R^2, that obtains the fit between the predicted and the real data (Eq. (6)).

$$R^2 = \left(\frac{cov(f, y)}{\sigma_f \sigma_y}\right)^2 \tag{6}$$

where $cov(f, y)$ represents the covariance between the model output and the real output, and σ_f and σ_y are the standard deviation of model output and the real output, respectively.

3.2. Data partitioning configuration

Most statistical procedures aimed to generate a prediction model, including RSM, are usually conducted in the same way. The parameters of a certain model are adjusted utilizing all the available data patterns. Then, the goodness of the model is assessed through some statistical indicators, such as R^2, over the same data. Commonly, the performance of the model against new data samples, never seen during the model generation process, is not evaluated a priori. This is a scenario quite different to that followed in the machine learning field, where the training of the model is done with only a subset of the available patterns, while the performance is assessed using a different subset. The goal is to test the generalization capability of the model, determining if it is able to produce a good solution even when data not seen before is given as input.

The way the patterns are partitioned into training and test subsets mostly depends on the amount of available data. There are three frequent approaches to follow:

Table 3
Results obtained from training data sets.

Data base	RSM		Reg-CO^{2}RBFN		FUZZY-GAP		MLP-BP		NU-SVM	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
GRL	0.7703	2.4133	**0.8160**	**2.1418**	0.0367	5.1966	0.0707	5.1249	0.0911	5.3366
GRS	**0.9582**	**2.8064**	0.9569	2.8321	0.5229	16.6859	0.6265	15.5074	0.7095	7.7735
HSRL	0.9024	7.7080	**0.9591**	**4.8921**	0.0933	26.4176	0.2515	23.2634	0.1549	22.7853
HSRS	0.9715	3.5590	**0.9854**	**2.5153**	0.3703	21.0612	0.6304	20.9095	0.5294	16.2654
SR	0.9909	0.9235	**0.9920**	**0.8540**	0.5464	17.0043	0.8627	4.0347	0.7097	5.4324
YEH	0.5472	5.7645	**0.8846**	**2.8440**	0.3187	7.8034	0.6807	5.4435	0.1067	9.6082
YWIS	0.9390	7.1809	**0.9919**	**2.5868**	0.5312	24.3770	0.9072	10.7019	0.7784	13.8560

- If a large amount of data is available, a random subset of patterns, usually between 50% and 70%, is used to train the model, while the remainder ones serve to evaluate it. This is commonly known as hold-out validation.
- When the number of patterns at disposal is not so large, the common approach is to perform a 5 or 10 folds cross validation. This consists in repeating the training/testing process 5 (10) times, using the 80% (90%) of data to build the model and the remainder 20% (10%) for testing it. The average of all iterations is given as final performance result, thus avoiding the potential bias that could be introduced by using hold-out.
- Sometimes the process to acquire new data patterns can be very difficult or expensive, resulting in that only a few of them are available. In order to increase the fairness on assessing the model, leave one out validation is usually applied in these cases. Leave one out is like cross validation using as many folds as samples there are in the data set. All but one pattern are used to build the model, while the remainder sample serves to compute the evaluation metric. The individual scores are eventually averaged, as in cross validation.

Since only 20 patterns have been used in this experimentation, the leave one out validation approach has been followed. This means that 20 models haven been generated and evaluated, each one of them using 19 patterns for training and the remainder one for testing. The reported performance indicators are average values from these 20 runs.

3.3. Obtained results

In Table 3 the results obtained with the training data sets are shown. For each method, the RMSE and the R^2 coefficient are calculated. Best results are in bold, the lowest value for RMSE and the highest for R^2.

As preliminary conclusion, in six out of seven variables Reg-CO^{2}RBFN outperforms the remaining methods both in RMSE error and R^2 coefficient. The RSM method achieves one best result for the GRS output variable, closely followed by Reg-CO^{2}RBFN. RSM obtains the second best result for five data sets, sometimes with results close to the best one, as in GRL or SR. In any case, it is remarkable that RSM achieves a low R^2 for GRL an YEH.

The other DM methods show mixed results for training data sets. Beginning with RMSE they obtain competitive results for GRL or YEH, even MLP-BP reaches the second best position for YEH. However, these DM methods show high RMSE values for the remaining data sets. Regarding the R^2 coefficients, it is difficult to find values close or higher than 0.9, only MLP-BP for SR or YWIS reaches this goal. NU-SVM obtain their best R^2 results, between 0.7 and 0.77, for GRS, SR and YWIS and poor values, below 0.16, for GRL, HSRL and YEH. The worst R^2 values are obtained by FUZZY-GAP with a maximum of 0.5464 and a minimum of 0.0367.

As mentioned, a relevant stage in this type of experiments is to test the models with unseen patterns or instances. The RMSE obtained by the models from the tests data sets is shown in Table 4, best results are in bold. The R^2 coefficient cannot be calculated for test data, as the leave one out methodology is used for data partitioning and test data sets are composed of only one instance.

For the test data sets, it must be highlighted that Reg-CO^{2}RBFN outperforms the other methods in five out of the seven variables: HSRL, HSRS, SR, YEH and YWIS. For GRL, FUZZY-GAP obtains one best result and Reg-CO^{2}RBFN the second best result. For GRS, RSM achieves the best result and Reg-CO^{2}RBFN the second one. RSM sometimes reaches second best result, but for GRL and YEH it appears as the worst performing method. In general, the remaining data mining methods have improved their accuracy, as mentioned for GRL and YEH, but for certain data sets, such as HSRL or HSRS, obtain poor results.

3.4. Graphical analysis of the results

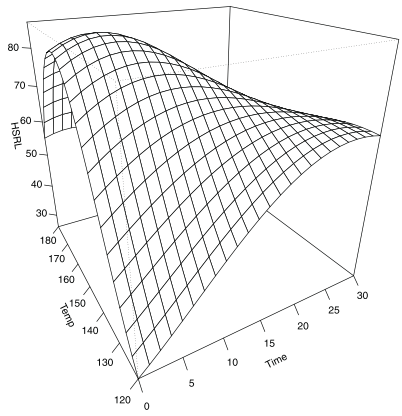
With the aim of carrying out a graphical analysis of the results, Figs. 2 and 3 show the output surface of four output variables (HSRL, SR, YEH and YWIS) corresponding to the models produced by Reg-CO^{2}RBFN and RSM, specifically for the first data partition. As we have three input variables, it is necessary to set one to a fixed value, representing the graphics the variations of the remaining two input variables. To achieve a better representation the fixed value is established to the central point of the range of the input variable.

This analysis starts with the models obtained for the HSRL variable (upper half in Fig. 2). If the shapes of these models are studied it can be concluded that Reg-CO^{2}RBFN can reach more complex contours. These contours are the results of the independent placement of gaussian function. On the other hand, RSM shapes come from second degree polynomials and may not always fit with precision any shape. If we observe the values achieved by the model, also we can conclude that Reg-CO2RBFN reproduce the training data more accurately. For this figure, $FeCl_3$ is fixed to 0.125. In this way as in the real values showed in Table 1, the model shows values around 80 when Time is 0, Temp is 160 and $FeCl_3$ is 0.125, or around 60 in the opposite when Time is 30. On the other hand, the values for the RSM model range around 60 for these examples.

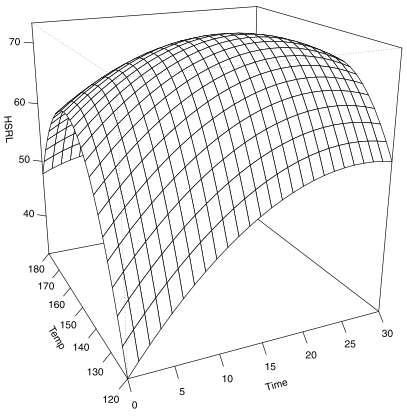
For the SR output variable (lower half in Fig. 2) both models obtain similar shapes, fixing Temp to 150. The maximum value for the variable, around 66, is achieved when Time is 0 and $FeCl_3$ is 0.05 and the minimum around 53 when Time is 0 and $FeCl_3$ is

Table 4
Results obtained from test data sets (RMSE).

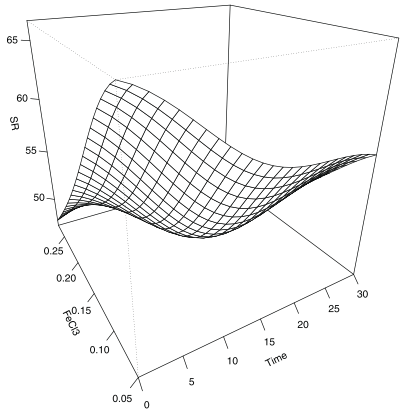
Data base	RSM	Reg-CO^{2}RBFN	FUZZY-GAP	MLP-BP	NU-SVM
GRL	5.5197	4.6436	**4.0923**	4.8251	4.8628
GRS	**5.4755**	6.2263	11.1294	15.1152	7.1436
HSRL	17.4745	**12.5074**	26.2649	21.9661	21.1395
HSRS	8.4328	**6.7960**	17.2009	18.6207	15.8235
SR	2.1124	**1.7823**	14.7954	4.2699	5.1285
YEH	12.1285	**6.5544**	7.0571	7.1525	9.3898
YWIS	15.1255	**7.1723**	21.2421	9.8838	12.4383



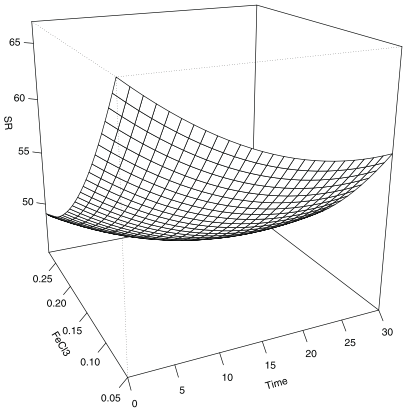
(a) HSRL vs Temperature/Time



(b) HSRL vs Temperature/Time



(c) SolidRecovery vs FeCl3/Time



(d) SolidRecovery vs FeCl3/Time

Fig. 2. Comparison among surfaces produced by Reg-CO^{2}RBFN (left) and RSM (right) models, HSRL and SR variables.

0.25. Nevertheless, Reg-CO^{2}RBFN shows more adaptive contours that RSM.

For the YWIS output (lower half in Fig. 3) the models obtained by Reg-CO^{2}RBFN and RSM are similar. For this figure the input variable Time was fixed to 15. As can be seen the value of YWIS reaches a value of 20 when Temp is set to 120 regardless of the value of $FeCl_3$. The maximum value is achieved when Temp is about 180 and $FeCl_3$ is 0.275. In any case the adaptive capacity of Reg-CO^{2}RBFN can be seen in the shapes this of model.

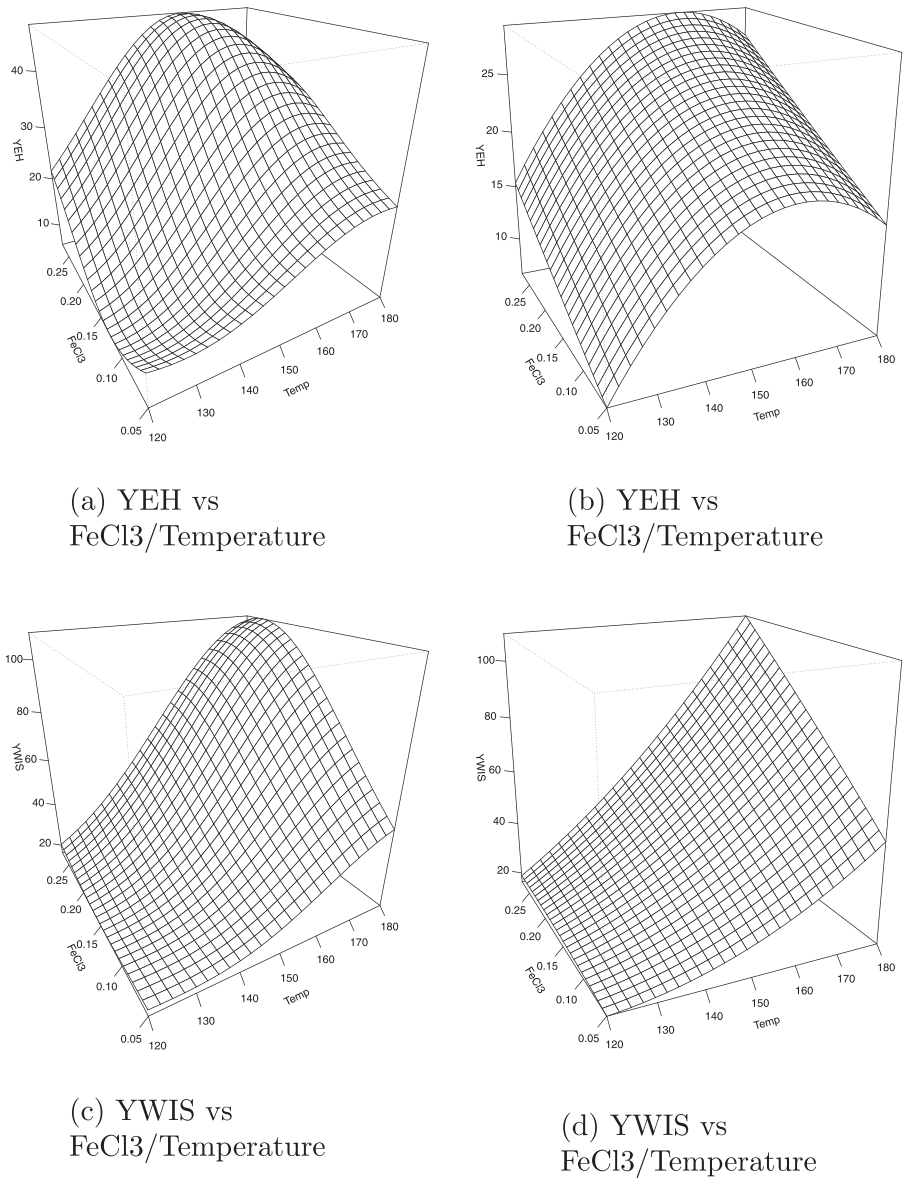
The models obtained by RSM and Reg-CO^{2}RBFN for YEH are the ones in the upper half of Fig. 3. While the RSM model shows a more uniform shape, Reg-CO^{2}RBFN model seems to fit better the training data. Taking into account that Time is fixed to 15 and analyzing maximum and values, it can be observed that when Temp is 120 and $FeCl_3$ is 0.05 Reg-CO^{2}RBFN returns a value around 12, similar to the corresponding training data, however, for these inputs the RSM model returns a value around 5. On the other hand, maximum real values, around 36, for YEH are observable in Reg-CO^{2}RBFN model when Temp is 160 and $FeCl_3$ is 0.2 Reg-CO^{2}RBFN, returning RSM model around 25 for these inputs.

3.5. Statistical analysis of the results

To achieve a more formal analysis of the results (statistically supported), hypothesis testing techniques (García et al., 2010; Sheskin, 2006) are used. In this field we have two options: parametric or nonparametric tests. To apply parametric test some data conditions, such as independency, normality and homocedasticity must be fulfilled. As these conditions are not guaranteed (Demšar, 2006), nonparametric tests haven been applied.

According to García et al. (2010), the first step is to detect statistical differences among a group of results, for example from methods applied to a given data set. For this goal, Friedman test (Sheskin, 2006) is used. This test establishes as null hypothesis that similar results, without significant differences, have been obtained by the methods. If the null hypothesis is rejected, the second step consists of applying a post-hoc test in order to determine the methods with significant differences with respect to the control algorithm or the algorithm with the best results. As post-hoc procedure, the Holm test (Holm, 1979) is used. For these tests, a p-value value associated with the validity of the null hypothesis is calculated. A p-value ranges from 0 to 1. When it is below a certain threshold α, it implies that a significant difference exists. Usually α is set to 0.05, which means that results are given with a 95% confidence level. Finally, as a general rule, this methodology is applied to the results obtained from the test data sets.

Firstly the Friedman test is applied. This test computes a ranking of the algorithms in the following way. For each data set the algorithms are classified with respect to its accuracy position, i.e., to the algorithm with the best accuracy the value 1 is assigned, to



(a) YEH vs FeCl3/Temperature

(b) YEH vs FeCl3/Temperature

(c) YWIS vs FeCl3/Temperature

(d) YWIS vs FeCl3/Temperature

Fig. 3. Comparison among surfaces produced by Reg-CO^{2}RBFN (left) and RSM (right) models, YEH and YWIS variables.

Table 5
Average rankings of the algorithms (Friedman test).

Algorithm	Ranking
Reg-CO^{2}RBFN	1.2857
RSM	3.0000
FUZZY-GAP	3.7143
MLP-BP	3.5714
SVM	3.4286

Table 6
Holm test.

i	Algorithm	p_{Holm}
1	FUZZY-GAP	0.016237
2	MLP-BP	0.020523
3	SVM	0.022460
4	RSM	0.042522

the algorithm with the second best accuracy the value 2 is assigned and so on. The final rank is the average of these ranking per data set.

Table 5 shows the average rankings of the algorithms for the test data sets. A lower value in the ranking represents a better algorithm. As can be seen Reg-CO^{2}RBFN is the best algorithm in the ranking, followed by the classical RSM methodology.

The Friedman statistic, distributed in accordance with a chi-square with 4 degrees of freedom, is 11.0857 and the p-value computed by Friedman Test is 0.0256. This value below 0.05 implies significant differences between the algorithms.

As the existence of significant differences are demonstrated the Holm test, a multiple comparison post-hoc procedure, is applied. The objective is to determine which are the methods that present statistical differences with respect to the best method, called control algorithm. The results of the Holm test are shown in Table 6. Reg-CO^{2}RBFN is the control algorithm (not appearing in the table) and the rest of methods are sorted by its p-value. The null hypothesis or the limit to establish significant differences (p-value) is shown in the this table.

As can be observed, all p-values are well below the usual $\alpha = 0.05$ value, therefore it can be affirmed that Reg-CO^{2}RBFN outperforms with significant differences the remaining methods. For a wider

description on the use of these tests, please refer to (Demšar, 2006; García et al., 2010).

4. Conclusions

In this paper the RSM method and different DM methods were tested to model the enzymatic hydrolysis process of OTB. The DM methods applied were Reg-CO^{2}RBFN, an RBFN design technique developed by the authors, an MLP, an SVM and a Fuzzy System. To guarantee the reliability of obtained results, a cross validation procedure is used and non-parametric statistical techniques supported the analysis carried out.

A first conclusion is that Reg-CO^{2}RBFN outperforms with statistical significant differences the remaining methods. The second best method is RSM. The other DM methods have obtained results with different accuracy, that is in large part attributable to the low number of available data patterns.

From the graphical analysis of the obtained models, we can conclude that Reg-CO^{2}RBFN models fit the data better than RSM models, accurately reproducing the training data. This is due to different capacity of the shapes generated from RSM methods, based on second degree polynomials, and from Reg-CO^{2}RBFN methods based on free placement of gaussian functions.

Further works will be focused on the use of this model under different operational conditions and pretreatment methods.

Acknowledgements

The authors are grateful to Spanish Ministerio de Economía y Competitividad (Ref. ENE2014-60090-C2-2-R and TIN2015-68454-R, including FEDER funds), for financing this research.

References

Affenzeller, M., 2009. Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. CRC Press.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F., 2011. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Multiple-Valued Logic Soft Comput. 17, 255–287, http://dx.doi.org/10.1007/s00500-008-0323-y.

Behera, S., Arora, R., Nandhagopal, N., Kumar, S., 2014. Importance of chemical pretreatment for bioconversion of lignocellulosic biomass. Renew. Sustain. Energy Rev. 36, 91–106, http://dx.doi.org/10.1016/j.rser.2014.04.047.

Bezerra, M.A., Santelli, R.E., Oliveira, E.P., Villar, L.S., Escaleira, L.A., 2008. Response surface methodology (RSM) as a tool for optimization in analytical chemistry. Talanta 76, 965–977, http://dx.doi.org/10.1016/j.talanta.2008.05.019.

Boser, B.E., Guyon, I.M., Vapnik, V.N.,1992. A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory – COLT '92. ACM Press, New York, NY, USA, pp. 144–152, http://dx.doi.org/10.1145/130385.130401.

Broomhead, D., Lowe, D., 1988. Multivariable functional interpolation and adaptive networks. Complex Syst. 2, 321–355.

Cao, W., Liu, Q., Wang, Y., Mujtaba, I., 2016. Modeling and simulation of VMD desalination process by ANN. Comput. Chem. Eng. 84, 96–103, http://dx.doi.org/10.1016/j.compchemeng.2015.08.019.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30.

Fan, R., Chen, P., Lin, C., 2005. Working set selection using the second order information for training SVM. J. Mach. Learn. Res. 6, 1889–1918.

García, S., Fernández, A., Luengo, J., Herrera, F., 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf. Sci. 180, 2044–2064.

Gaudart, J., Giusiano, B., Huiart, L., 2004. Comparison of the performance of multi-layer perceptron and linear regression for epidemiological data. Comput. Stat. Data Anal. 44, 547–570, http://dx.doi.org/10.1016/S0167-9473(02)00257-8.

Harris, C.J., 1989. Fuzzy sets. Inf. Control 2, 267–285.

Holm, S., 1979. A simple sequentially rejective multiple test procedure. Scand. J. Stat. 6, 65–70.

Howard, L., D'Angelo, D., 1995. The GA-P: a genetic algorithm and genetic programming hybrid. IEEE Expert 10, 11–15, http://dx.doi.org/10.1109/64.393137.

López-Linares, J., Romero, I., Moya, M., Cara, C., Ruiz, E., Castro, E., 2013. Pretreatment of olive tree biomass with fecl3 prior enzymatic hydrolysis. Bioresour. Technol. 128, 180–187, http://dx.doi.org/10.1016/j.biortech.2012.10.076.

Maimon, O., Rokach, L., 2010. The Data Mining and Knowledge Discovery Handbook, 2nd ed. Springer.

Ochoa-Estopier, L., Jobson, M., Smith, R., 2013. Operational optimization of crude oil distillation systems using artificial neural networks. Comput. Chem. Eng. 59, 178–185, http://dx.doi.org/10.1016/j.compchemeng.2013.05.030.

Park, J., Sandberg, I., 1993. Universal approximation and radial basis function network. Neural Comput. 5, 305–316.

Pérez-Godoy, M., Rivera, A., del Jesus, M., Berlanga, F., 2010. CO^{2}RBFN: an evolutionary cooperative-competitive RBFN design algorithm for classification problems. Soft Comput. 14, 953–971.

Platt, J., 1998. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.

Ravindran, R., Jaiswal, A.K., 2016. A comprehensive review on pre-treatment strategy for lignocellulosic food industry waste: challenges and opportunities. Bioresour. Technol. 199, 92–102, http://dx.doi.org/10.1016/j.biortech.2015.07.106.

Rojas, R., 1996. Neural Networks. Springer Berlin Heidelberg, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-642-61068-4.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1985. Learning internal representations by error propagation. Technical Report. DTIC Document.

Sánchez, L., Couso, I., 2000. Fuzzy random variables-based modeling with GA-P algorithms. In: Bouchon, B., Yager, R.R., Zadeh, L. (Eds.), Information, Uncertainty and Fusion. , pp. 245–256.

Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L., 2000. New support vector Algorithms. Neural Comput. 12, 1207–1245, http://dx.doi.org/10.1162/089976600300015565.

Sheskin, D., 2006. Handbook of Parametric and Nonparametric Statistical Procedures, 2nd ed. Chapman & Hall/CRC.

Vapnik, V., Vapnik, V., Golowich, S.E., Smola, A., 1996. Support vector method for function approximation, regression estimation, and signal processing. Adv. Neural Inf. Process. Syst. 9, 281–287.

Wang, Y., Cao, H., Yan, X., Zhou, Y., Liu, X., McLoone, S., 2016. A generalized fuzzy linguistic model for predicting component concentrations in an optical gas sensing system. Chemometr. Intell. Lab. Syst. 158, 21–30, http://dx.doi.org/10.1016/j.chemolab.2016.07.012.

Zhao, Y., Zhang, X., Deng, L., Zhang, S., 2016. Prediction of viscosity of imidazolium-based ionic liquids using MLR and SVM algorithms. Comput. Chem. Eng. 92, 37–42, http://dx.doi.org/10.1016/j.compchemeng.2016.04.035.