Design and Application of Soft Sensor Using Ensemble Methods

Symone Soares¹, Rui Araújo¹, Pedro Sousa¹,², and Francisco Souza¹

¹Institute of Systems and Robotics (ISR-UC), and Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal.

²Acontrol - Automation and Industrial Control, Ltd, Parque Empresarial de Eiras, Lote 5, PT-3020-265 Coimbra, Portugal.
symonesoares@isr.uc.pt, rui@isr.uc.pt, pasousa@acontrol.pt and fasouza@isr.uc.pt

Abstract

Industries are faced with the choice of suitable process control policies to improve costs, quality and raw material consumption. In the paper pulp industry, it is important to estimate quickly the Chemical Oxygen Demand (COD), a parameter that is highly correlated to product quality. Soft Sensors (SSs) have been established as alternative to hardware sensors and laboratory measurements for monitoring and control purposes. However, in real setups it is often difficult to get sufficient data for SS development. This work proposes Ensemble Methods (EMs) as a way to improve the SS performance for small datasets. EMs use a set of models to obtain better prediction. Their success is usually attributed to the diversity. Bootstrap and noise injection are used to produce diverse models. Several combinations of EMs are compared. The SS is successfully applied to estimate COD in a pulp process.

1. Introduction

Two decades ago researchers started to make use of the large amounts of data being measured and stored in the process industry by building predictive models. In the context of process industry, these models are called Soft Sensors (SSs) [19]. SSs are a valuable tool in many industrial fields of application, such as, pulp and paper mill [14], wastewater treatment systems [16], petrochemical [26] and cement kiln [27].

According to [11], the choice of SSs has been strengthened by the need of suitable production policies to improve final product prices, quality and raw material consumption. For example, in the paper pulp industry, it is important to estimate quickly the Chemical Oxygen Demand (COD), a parameter that is highly correlated to the quality of paper pulp process. High value of COD indicates poor washing of paper and requires more chemicals.

SSs use computational intelligence methodologies for online estimation of variables which cannot be automatically measured at all, or can only be measured at high cost, inaccurately, sporadically or with high delays (e.g. laboratory analysis) [21]. They may bring information about the process state which may be extracted from observing a group of relevant measurements. Another benefit is the delivery of additional information about process features, which cannot be measured using sensors and have to be evaluated with high delay [18].

Unfortunately there are some unaddressed issues on the SS development and maintenance [19]. The performance of SSs relies on the quality of the data that are used to extract knowledge during the identification procedure. In some cases, the databases are small and require very long periods of time to be significantly enlarged. Small datasets offer some drawbacks because they have insufficient information about processes.

From the Machine Learning point of view, the amount of collected data significantly influences the accuracy of learning. This is because the model may adjust to specific random features of the training data, that have no causal relation to the target function. In this process, called overfitting, the performance on the training samples still increases while the performance on unseen data becomes worse.

According to [2], in Neural Network (NN) learning, the strategies to avoid overfitting can be divided into strategies applied before learning and strategies applied after learning. In strategies applied before learning, training samples are pre-processed or new samples are artificially created, for example, by noise injection [15] or bootstrap [10]. Noise injection creates artificial datasets by adding some noise to the original dataset. Bootstrap draws multiple datasets by random sampling with replacement from the original dataset. Strategies applied after learning include weight decay [20], regularization methods [1], Ensemble Methods (EMs) [23] and pruning techniques for elimination of nodes and connections from the trained network [3].
EMs have been proved to improve the model’s prediction performance in the cases where only a small amount of data is available. A set of base models is trained and their responses are combined in order to obtain the final prediction [9]. The key to the success of EMs is the diversity. Popular techniques to achieve diversity can be given by starting the learning of models with different initial conditions, manipulating the training dataset, using different architectures and learning algorithms.

The second key component of an EM is the strategy employed to combine models. Combining a set of models can be thought of as a way of managing the recognized limitations and diversity of the individual models.

This work compares different strategies for SS modeling when a small dataset is available. The main focus is to build EMs of NNs to improve the SS generalization capability. The adopted techniques to achieve diversity on the EMs are bootstrap and noise injection. The paper also proposes and tests different strategies of combination to create robust EMs for SS design. The SS was successfully applied in a Paper Pulp Industry for estimation of the COD.

The paper is organized as follows. In Section 2, a description of the Paper Pulp Process is reported. Section 3, discusses some aspects of Ensemble Methods. Techniques for dataset manipulation are presented in Section 4. In Section 5 several approaches to combine Ensembles are presented. Section 6, experiments to estimate the COD are reported. Section 7 contains some concluding remarks.

# 2 Paper Pulp Process

The paper pulp process converts wood chips into clean pulp with uniform quality, where wood fibers are separated from each other. The fibers are joined by dissolving the lignin (60-90%) with chemicals that leave the fibers relatively unattached. The result of the process will feed the paper machines, where it is desirable to have pulp fibers with consistent strength, drainage characteristics, and wet end chemistry properties.

This work is performed using a plant based on the Kraft Pulping Process. Kraft consists of producing pulp by cooking wood chips in the presence of an alkaline solution (it has a high pH) during which time the chemicals react with the lignin in the wood [7]. The stages of the process are shown in the Table 1, where SWOD means Screening, Washing and Oxygen Delignification.

<table>
<thead>
<tr>
<th>Wood Preparation</th>
<th>The wood is prepared for the Digester, with uniform thick chips (2-6mm) containing a small percentage of bark. Particles with size deviation will not be fully penetrated by the liquor Digester. Softwood and hardwood will be segregated, because it is difficult to maintain uniform cooking of them due to their different reaction rates.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digester</td>
<td>It is a chemical reactor in which a mixture of wood chips and liquor is heated by steam. The aim is to separate the lignin from the cellulose fibers by chemical reactions (called delignification). The lignin is dissolved by the process with small reactions between the dissolvant and the cellulose fibers. The liquor is an alkaline solution consisting mainly of sodium hydroxide and sodium sulfite in water.</td>
</tr>
<tr>
<td>SWOD</td>
<td>The pulp is filtered and washed. The goal is to remove solid impurities from acceptable pulp fibers. Impurities differ in/or specific gravity from fibers. This facilitates their removal by perforated or slotted plates in various types of screens or equipment using gravity or centrifugal forces. Finally some lignin, which is not dissolved in the digester, is separated from the fibers by processing it in an oxygen reaction.</td>
</tr>
<tr>
<td>Bleaching Plant</td>
<td>Bleaching is the chemical processing to decrease the color of the pulp, so that it becomes whiter. Whiteness is an important quality factor.</td>
</tr>
</tbody>
</table>

The aim of the process is to maximize the pulp quality and yield, reduce the overall operating cost including energy usage, and minimize the adverse environmental impacts of pulp mills. The most important variable in the paper pulp process is the COD. Low quantity of COD carry over to the delignification and bleaching stages indicates good washing and requires less chemicals.

The conventional dichromate method for COD determination has several drawbacks. It is a time consuming (2-4 hours) and expensive reflux process to allow for complete oxidation of the organic material. To address the challenge that the primary variable COD is not measured online, the Soft Sensor have been designed to estimate the COD based on the secondary variables.

# 3 Ensemble Methods

Over the last decade, Ensemble Methods (EMs) have enjoyed a growing attention and popularity due to their many desired properties [23]. EMs combine multiple models to produce robust predictions. This aggregation of models is generally more accurate than any of the individual models making up the EM [9]. Two popular methods to create EMs are Bagging [4] and Boosting [13]. These methods rely on resampling techniques.

Both theoretical and empirical research has demonstrated that a good EM is one where the individual models are both accurate and diverse [22]. An accurate model is one that has a low error rate for new data values; and two models are diverse if they make different error on new data points.

In Machine Learning, for single models, the bias-variance decomposition for quadratic loss states that the generalization error of an estimator can be broken into bias and variance. They usually work in opposition to each other. The bias can be characterized as a measure of
how close, on average over several different training sets, your estimator is to its target. The variance is a measure of how “stable” the solution is: given slightly different training data, an estimator with high variance will tend to produce wildly varying performance. Training an estimator for a long time tends to decrease bias, but slowly increase variance; at some point there will be an optimal trade-off that minimizes the generalization error [5].

This concept is extended to take account of the possibility that the estimator could be an EM of Estimators. The bias-variance-covariance decomposition states that the squared error of EM can be broken into three terms, bias, variance and covariance [25]. Let denote a dataset \( z = \{(x_i, y_i)\}_{i=1}^n \) of size \( n \), where \( x_i \in \mathbb{R}^{v \times 1} \) ( \( v \) is the number of input variables) and \( y_i \) is an output. Consider each element drawn from a random variable \( Z \) defined over an unknown distribution \( p(x, y) \). Let denote \( f_1, \ldots, f_N \) a collection of \( N \) estimators. We redefine our random variable \( Z \) as a set \( Z = (Z_1, \ldots, Z_N) \), so the \( i \)th estimator is trained with a training set \( z_i \) drawn from its own random variable \( Z_i \).

Let consider \( E\{\cdot\} \) the expectation with respect to the dataset that is used to train a model \( f_i \). Finally, the bias-variance-covariance decomposition is expressed as follows:

\[
E[(f - y)^2] = \text{bias}^2 + \frac{1}{N} \text{var} + \left(1 - \frac{1}{N}\right) \text{covar},
\]

\[
\text{bias} = \frac{1}{N} \sum_i E\{f_i\} - y,
\]

\[
\text{var} = \frac{1}{N} \sum_i E\{(f_i - E\{f_i\})^2\},
\]

\[
\text{covar} = \frac{1}{N(N-1)} \sum_{i,j;i\neq j} E\{*,\},
\]

\[
E\{*,\} = (f_i - E\{f_i\})(f_j - E\{f_j\}),
\]

and \( \bar{f} = \frac{1}{N} \sum_{i=1}^N f_i \), where \( \bar{f} \) is the final output of EM.

This shows that the generalization error of an EM not only depends on the bias and variance of the EM’s models, but also on the relationships between individual members of the EM, quantified in the covariance term. This term can be said to indicate the diversity between EM’s models as far as their error estimates are concerned. It is believed that the more diverse models an EM has, the less correlated they would be. The lower the covariance term, the less the error correlation amongst the networks, which implies reduced error and better performance at the EM. This is the main reason why diversity in EMs is extremely important [8].

According to [6], an EM can be constructed by introducing diversity in three main ways:

1. By making the component models start the learning process with different initial conditions. For example, different initialization of weights for Neural Networks;

2. By manipulating the training dataset (such as bootstrap and noise injection); using different architectures or different learning algorithms.

3. By modifying the trajectory used by the components in the search space using penalty function (such as a regularization term) methods and global search methods (such as evolutionary algorithms).

This work adopts approaches 1 and 2 to introduce diversity to Neural Network models, e.g. by using different initialization of weights, different architectures and different training datasets with bootstrap and noise injection.

Optimal combination of models is a way to create accurate EMs [23]. Combining a set of models is a way of managing the recognized limitations and diversity of the models in EMs. A comparison of approaches to combine EMs is carried out in this paper.

4 Manipulating the Dataset

Experimental results indicate that the manipulation of datasets can improve the performance of EMs when the number of available data points is limited. Below, two approaches are investigated: bootstrap and noise injection.

4.1 Bootstrap

Bootstrap draws datasets by simple random sampling with replacement from the original dataset [10]. Bootstrap can be used to expand upon a single realization of a distribution or to create a set of bootstrapped datasets that can provide a better understanding of the average and variability of the original unknown distribution.

Bootstrap resampling has been applied to form a Bootstrap Aggregation (e.g. Bagging) [4]. Bagging creates \( N \) Neural Networks (NNs) with independent training and using different architectures and different training datasets, which are generated by forming bootstrap replicates of the original training dataset.

Suppose a dataset \( D = \{(x_i, y_i)\}_{i=1}^n \) of size \( n \), where \( x_i \in \mathbb{R}^{v \times 1} \) ( \( v \) is the number of input variables) and \( y_i \) is an output. A bootstrap dataset \( D^* = \{(x_i^*, y_i^*)\}_{i=1}^n \) is constructed by random drawing \( n \) times with replacement from dataset \( D \).

Considering that \( B \) datasets are generated, where \( b \) denotes a dataset, \( b = 1, 2, \ldots, B \). For each dataset \( b \) a NN model is trained. An individual model fitted with \( b \) is given by \( f_b(x) \). The estimation of the Bagging of NNs is calculated as the average of individual NNs:

\[
\bar{f}(x) = \frac{1}{B} \sum_{b=1}^B f_b(x)
\]
4.2 Noise Injection

In noise injection, virtual data is produced by adding noise to the original data. Adding noise during the learning process can enhance the generalization capability [15]. This is because noise helps to prevent overfitting by providing additional constraints.

For NNs, results indicate that noise injection can reduce overfitting to a greater degree than methods like early stopping and weight decay [17]. For EMs, the motivation is that noise injection can increase the diversity. In [28] an EM is proposed to improve forecasting performance by adding noises to the input data to form models based on different training samples. Moreover, [12] and [24] propose noise injection and bootstrap to increase the diversity of the EMs.

The issue of appropriately choosing noise has not been completely addressed in the literature. This choice must be problem dependent as different data has different inherent noise levels. However, it is clear that large noises could distort the underlying feature, while smaller noises may not have enough impact [28].

Two methodologies of noise injection can be performed: incremental (i.e. noise added to samples in each iteration of training) and non-incremental (i.e. noise added to samples before training). Our purpose is to investigate non-incremental noise injection to create several training datasets with the same size of the original dataset, where each new dataset is used to train an individual NN model in the EM.

Three organizations of the operations with noise injection will be analyzed: noise injection into inputs, noise injection into outputs and noise injection into both inputs and outputs.

Assume a dataset $D = \{(x_i, y_i)\}_{i=1}^n$ consisting of $n$ samples, where $x_i \in \mathbb{R}^{v \times 1}$ ($v$ is the number of input variables) and $y_i$ is an output. A matrix $X \in \mathbb{R}^{n \times v}$ and an output vector $y \in \mathbb{R}^{n \times 1}$ are created, where the input vectors $x_i$ are arranged in the rows of $X$.

Random zero mean Gaussian noise vectors with distribution $N(0, \sigma^2)$ and fixed variance are added to original samples. Assume a noise vector denoted as $u = [u_1, \ldots, u_v]^T$, where $u \in \mathbb{R}^{n \times 1}$.

For noise injection into inputs, a noise version of input matrix is given by:

$$X' = X + U_{\text{input}},$$

where $U_{\text{input}} = [u_1 \ldots u_v]$.

For noise injection into outputs, a noise version of output vector is given by:

$$y' = y + u.$$

In this case, no noise is added to the matrix of input data. Then the dataset is represented by $D'_{\text{out}} = \{(x_i', y_i')\}_{i=1}^n$, where $y_i'$ is row $i$ of $y'$, for $i = 1, \ldots, n$.

When noise is injected into both inputs and outputs, the dataset is given by $D'_{\text{in-out}} = \{(x_i', y_i')\}_{i=1}^n$, where $x_i'$ is row $i$ of $X'$, and $y_i'$ is row $i$ of $y'$, for $i = 1, \ldots, n$.

5 Combining Ensemble Methods

Aggregation strategies are the second important operation involved in an EM. Optimal combination can enhance the robustness and accuracy of an EM. In the literature several approaches can be found for combining multiple models [23]. In this work, five different combination strategies are investigated to achieve good performance in EMs.

Consider a dataset $D = \{(x_i, y_i)\}_{i=1}^n$ of size $n$, where $x_i$ represents the input variables $(x_{i1}, y_{i2})$ and $y_i$ is the output variable; the proposed combinations are:

1. **Simple Mean**: The final output of EM is obtained as the average of all models’ outputs. Considering $N$ the number of models and $f_i(x)$ the prediction of each model based on input data $x$, the output of EM is given by:

$$\bar{f}(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x).$$

2. **Trimmed Mean**: The EM’s output is calculated as the trimmed mean of all models’ outputs. The aim is to remove the most pessimistic and optimistic models before calculating the mean, avoiding extreme values of models’ outputs. For $P\%$ trimmed mean, the mean is obtained by excluding $P\%/2$ from lowest models’ outputs and $P\%/2$ from highest models’ outputs.

3. **Weighted Average based on the Accuracy**: It is performed by taking a weighted sum of the output of each model, where the weight is based on the accuracy of each model. Here the accuracy of an individual model is based on the performance in the training, given by Correlation Coefficient (CC) between estimated output and real output. The mechanism of determination of individual weights $w_i$ and EM’s output is given below:

$$w_i = \frac{CC_i}{\sum_{i=1}^{N} CC_i},$$

$$\bar{f}(x) = \sum_{i=1}^{N} w_i \cdot f_i(x).$$
4. **Partial Least Squares Regression (PLSR):** A linear combination is obtained through PLSR of the models’ outputs. The coefficients of the PLSR are obtained by using the training dataset. The number of latent variables is determined using cross-validation.

5. **Neural Networks:** It is a nonlinear combination, where a NN receives as inputs the individual models’ outputs and the EM’s output is the NN’s output. In this paper, a Multilayer Perceptron (MLP) with one hidden layer is used for combination, and a growing strategy is performed to determine the number of neurons in the hidden layer. The training of the MLP used for combination is performed with the training dataset using the outputs of the individual models as inputs.

6. **Main Results**

   A small amount of samples offers some drawbacks to SS modeling, such as low generalization capability. Concerning this problem, it will be shown that improvements in generalization capability can be attained by introducing diversity in the EMs.

   A number of strategies will be used to increase diversity. These methods are developed by starting the learning of models with different conditions (i.e. initialization of weights); different architectures (i.e. by varying the number of neurons in the hidden layer) and manipulating the training dataset (i.e. bootstrap and noise injection). Moreover, it will be tested different types of combinations to balance the diversity of EMs.

   As described above, the SS was designed to estimate the COD in a Paper Pulp Industry. The acquired dataset was pre-processed to exclude outliers and select input variables. At the end of these processes, the dataset was composed of 506 samples and 13 input variables. For the learning step, the dataset was randomly divided into training dataset (80% of the samples) and testing dataset (20% of the samples).

   Here, the EMs are implemented with models using Multilayer Perceptron Neural Network (MLP NN), trained by the Levenberg-Marquardt algorithm. The architecture has only one hidden layer. And the performance of EMs are evaluated by Mean Square Error (MSE) and Coefficient Correlation (CC) between the estimated output and the real output in the testing dataset.

   The proposed method to build EMs of NNs is described as follows:

1. Generate \(N\) copies from the original training dataset;
2. Manipulate the \(N\) training datasets by applying bootstrap and/or noise injection;
3. Train a number of NNs for each training dataset and select the best NN based on the performance; The selected NN will be included in the EM;
4. Choose the best aggregation strategy for the NNs in the EM.

   The experiments presented in this paper are based on fifteen NNs, i.e. fifteen copies from the original training dataset. The second step aims to manipulate the \(N\) datasets. Then, for each perturbed training dataset, a NN is chosen among a number of NNs obtained by varying the number of neurons in the hidden layer (from 0 to 30), using two activation functions (linear and hyperbolic tangent) and three different initialization of weights. At the end of these steps, fifteen accurate NNs are obtained. And then the best combination method for the fifteen NNs is chosen. Five types of combination are analyzed: simple mean, 10% trimmed mean, weighted average based on the accuracy, PLS and NN.

   **6.1 Experiment 1 - Ensembles with Clean Dataset and Ensembles with Bootstrap**

   The goal of this experiment is to analyze the improvements of EMs with bootstrap compared to EMs with clean datasets. For EM with a clean dataset, step 2 described above is not executed, i.e. the dataset is not manipulated. Bootstrap repeatedly samples with replacement of the original training dataset. In both cases, an EM with fifteen NNs were generated and five types of combination are analyzed in this experiment.

   Table 2 displays the results of EMs for the different types of combinations. The results with best MSE and CC are in bold. Figure 1 depicts the real COD and the estimated COD by the best EM with bootstrap based on the CC criterion in the testing dataset. As can be seen, when CC is used as performance metric, the EM with bootstrap is more accurate than the EM with clean dataset; And mean, trimmed mean and weighted average combinations all have superior performance when compared with other types of combination.
6.2 Experiment 2 - Ensembles with Noise Injection

The second approach is to train all NNs of the EM according to three cases: noise injection into the (I) inputs, (II) outputs, (III) and both the inputs and outputs. This means that random vectors are added to the training dataset before they are presented to the NNs. Zero-mean Gaussian white-noise vectors with variance $\sigma^2$ are used. As discussed above, the issue of determining the best value of variance (a parameter that represents the “noise level” on the dataset) is an unaddressed question in the literature. So the objective is to experiment several levels of variance for analyze the resulting performances of the EMs.

The following three levels of variance $\sigma^2$ were tested: 0.01, 0.02 and 0.03. These values were chosen in order to simulate low, medium and high levels of noises. The above described four-step method is used to build an EM for each level of noise and each of the cases (I), (II), and (III). Tables 3, 4 and 5 show the main results of EMs for cases (I), (II) and (III), respectively. The best results, for each type of combination and metric of error, are in bold.

Table 3. Case I - Main Results of Ensembles with Noise Injection into Inputs.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Error</th>
<th>Variance $\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>MSE</td>
</tr>
<tr>
<td>Mean</td>
<td>0.6508</td>
<td>0.0107</td>
</tr>
<tr>
<td>Trained Mean</td>
<td>0.6350</td>
<td>0.0111</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>0.6476</td>
<td>0.0107</td>
</tr>
<tr>
<td>PLS</td>
<td>0.5678</td>
<td>0.0135</td>
</tr>
<tr>
<td>NN</td>
<td>0.5745</td>
<td>0.0127</td>
</tr>
</tbody>
</table>

As can be observed, EMs with noise injection into outputs are significantly more accurate than EM with clean dataset and EM with bootstrap resampling.

6.3 Experiment 3 - Ensembles with Both Bootstrap and Noise Injection

Here, the aim is to analyze the hybridization of two techniques of dataset manipulation: bootstrap resampling and noise injection. In this test, bootstrap was first applied and then noise injection into outputs with variance $\sigma^2 = 0.02$ was performed.

Table 6 shows the produced results. Trimmed mean obtained the best generalization capability when compared with the other aggregation strategies. The experiment indicates that EM with bootstrap and noise injection has better performance than EM with clean dataset and EM with
Table 5. Case III - Main Results of Ensembles with Noise Injection into Both Inputs and Outputs.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Error</th>
<th>Variance $\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>Mean</td>
<td>CC</td>
<td>0.6860</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0097</td>
</tr>
<tr>
<td>Trained Mean</td>
<td>CC</td>
<td>0.6736</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0101</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>CC</td>
<td>0.6894</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0095</td>
</tr>
<tr>
<td>PLS</td>
<td>CC</td>
<td>0.6787</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0109</td>
</tr>
<tr>
<td>NN</td>
<td>CC</td>
<td>0.6903</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0097</td>
</tr>
</tbody>
</table>

Table 6. Main Results of Ensembles with Both Bootstrap and Noise Injection.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Error</th>
<th>Both Bootstrap and Noise Injection (II) with $\sigma^2$=0.02</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.7100</td>
</tr>
<tr>
<td>Mean</td>
<td>CC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0100</td>
</tr>
<tr>
<td>Trained Mean</td>
<td>CC</td>
<td>0.7168</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0095</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>CC</td>
<td>0.7042</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0109</td>
</tr>
<tr>
<td>PLS</td>
<td>CC</td>
<td>0.6762</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0193</td>
</tr>
<tr>
<td>NN</td>
<td>CC</td>
<td>0.6812</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>0.0142</td>
</tr>
</tbody>
</table>

bootstrap. However, it has worse performance than EM with noise injection into outputs, i.e. case (II).

7 Conclusions

In the Paper Pulp, it is important to estimate quickly the COD, a parameter that is highly correlated to product quality. Conventional methods for COD determination usually are time consuming (2-4 hours) and expensive. To cover these drawbacks, a Soft Sensor was designed to estimate COD.

Unfortunately, the size of dataset that was used for Soft Sensor modeling is small and it has insufficient information about paper pulp process. From the Machine Learning point of view, the amount of collected data significantly influences the accuracy of learning.

To address this challenge, the paper compared a number of strategies to improve the generalization capability of Soft Sensors based on Ensemble Methods when only a small amount of data is available. The aim of the proposed techniques is to improve the accuracy and diversity of the Ensemble’s models by manipulating the dataset with bootstrap and noise injection. Moreover, different types of combinations were experimented to improve the performance of Ensembles.

Results indicate that the construction of Ensembles with noise injection into outputs have superior performance compared to other Ensembles (e.g. Ensembles with bootstrap, Ensembles with noise injection into inputs, Ensembles with noise injection into both inputs and outputs, and Ensembles with both bootstrap and noise injection).

Furthermore, the tests showed that simple mean, trimmed mean and weighted average on the accuracy have better performance when compared with other types of combination, such as, PLS and Neural Network combinations.

The good results obtained with the developed methodologies allowed the estimation of Chemical Oxygen Demand (COD) in a Pulp Industry process. Future works include the development of new techniques for dataset manipulation and for combination of models in Ensemble Methods.

Acknowledgment

This work was supported by Mais Centro Operacional Program, financed by European Regional Development Fund (ERDF), and Agência de Inovação (AdI) under Project SInCACI/3120/2009.
Symone Soares and Francisco Souza have been supported by Fundação para a Ciência a Tecnologia (FCT) under grants SFRH/BD/68515/2010 and SFRH/BD/63454/2009, respectively.

References


