Review of Soft Sensors Methods for Regression Applications

Francisco A. A. Souza[a,b,*], Rui Araújo[a,b], Jérôme Mendes[a,b]

[a] *DEEC-Department of Electrical and Computer Engineering; University of Coimbra, Pólo II; Coimbra, Portugal*
[b] *ISR-Institute of Systems and Robotics; University of Coimbra, Pólo II; Coimbra, Portugal*

Abstract

Soft sensors for regression applications (SSR) are inferential models that use on-line available sensors (e.g. temperature, pressure, flow rate, etc) to predict quality variables which cannot be automatically measured at all, or can only be measured at high cost, sporadically, or with high delays (e.g. laboratory analysis). SSR are built using historical data of the process, usually provided from the supervisory control and data acquisition (SCADA) system or obtained from laboratory annotations/measurements. In the SSR development, there are many issues to deal with. The main issues are the treatment of missing data, outlier detection, selection of input variables, model training, validation, and SSR maintenance. In this work a literature review, on each of these topics will be performed, reviewing the most important works in these areas. Emphasis will be given to the methods and not the applications.

Keywords: review, soft sensor, regression, prediction, chemometrics

1. Introduction

Industrial processes are well equipped with a variety of sensors, such as temperature, flow rate and pressure sensors, designed for online supervision, monitoring and control, and to maintain consistent product quality. Some variables, which may be quality variables for example, cannot be automatically measured online, due to the lack of sensors, or due to the high cost of the sensor, thus leading to the lack of enough information about the system state in real-time. Usually, laboratory tests of product samples are conducted to measure off-line the product quality on a specified interval base. In order to measure the quality variables in real-time, one can use computational intelligence methodologies to build intelligent/computational sensors to infer the value or the quality target variables from other on-line measured process variables. The basis for building such intelligent sensors is that the values of target variables, or the product quality, have a functional relationship with other process variables that can be measured on-line. Such kind of intelligent sensors is one of the applications of soft sensors [1; 2], refereed here as soft sensor for regression applications (SSR). They are important tools for many industrial processes, such as pulp and paper mills, wastewater treatment systems, cement kilns, refineries, and polymerization processes, just to give a few examples. In general terms, soft sensors can be defined as inferential models that use online available sensor measurements (easy to measure variables) for on-line estimation of quality variables (hard to measure variable) which cannot be automatically measured at all, or can only be measured at high cost, sporadically, or with high delays (e.g. laboratory analysis).

A SSR is a regression model which uses easy-to-measure variables to predict a hard-to-measure variable. It is subject of research in many areas. Originally, SSRs were studied as part of chemometrics, which stands for statistical methods for extracting information from data sets that often consist of many measured variables [3]. According to Wold [3]: "Chemometrics, is heavily dependent on the use of different kinds of mathematical models (high information models, ad hoc models, and analogy models). This task demands knowledge of statistics, numerical analysis, operation analysis, etc., and in all, applied mathematics.", i.e. chemometrics is not an isolated/sole research area. From the chemometrics literature it is possible to see the use of different approaches including machine learning and pattern recognition [4], artificial intelligence [5], system identification [6], and statistical learning theory [7]. Despite the fact that the objectives and emphasis on all these areas are different, they are intrinsically connected by the necessity to learn models from data. This point of view is further justified in the work done in [8], where the author revises the problem of system identification.

Then, the state of the art discussed here will not be limited the chemometrics literature, it will also discuss the main and recent contributions from the other areas.

SSR development encompasses the same design cycle of classical regression systems [9; 6]. However, it has its own

[*] Corresponding author at: Institute of Systems and Robotics (ISR-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal. Tel.: +351 910942012.

Email addresses: fasouza@isr.uc.pt, alexandre.andry@gmail.com (Francisco A. A. Souza), rui@isr.uc.pt (Rui Araújo), jermendes@isr.uc.pt (Jérôme Mendes)

peculiarities. SSR development has the following main steps [1; 2]: (I) data collection and filtering, (II) selection of input variables, (III) model choice and training, (IV) model validation, and (V) model maintenance. In the first stage the data is collected, and the goals of this stage include the handling of missing data and outliers. The goals of the second stage are the selection of most relevant inputs, and possibly also the respective time lags. The model choice and training requires the correct selection and learning of the model. The model validation step is necessary to judge if the learned model reproduces the target variables within acceptable quality or performance levels. The last step is SSR maintenance, where the goal is to maintain a good SSR response under the presence of process variations or some data change.

2. Data Collection and Pre-Processing

Industries are usually required to store their data from the processes. This is the basis for the subsequent use of such data for system optimization, or other related data driven methods. Unfortunately, data collection in real industrial applications comes with well know problems to deal with, such as problems with sampling time, missing data, outliers, working conditions, accuracy, and so on.

2.1. Sampling Time

In industrial systems some variables are acquired at different time rates. This is most evident when analyzing the sample rates of easy-to-measure and hard-to-measure variables. In the majority of problems the acquisition frequency of easy-to-measure variables is much higher than the acquisition frequency of hard-to-measure variables. In such cases there is the necessity to synchronize the variables. This problem is usually refereed in literature as multirate character, or multiple-rate phenomenon [10]. In practice the following two approaches are most commonly adopted:

1. Down-sample of the easy-to-measure data samples, in accordance with the slow sampling rate of the hard-to-measure variables, by excluding the samples of the easy-to-measure variables that do not have a corresponding hard-to-measure (target) value [11; 12];
2. Instead of excluding the samples that do not have the respective target, a finite impulse response (FIR) model is estimated and applied on the samples in order to estimate the hard-to-measure, low sampling rate, variables. The big concern in this approach is the selection of weighting values and length of the FIR filter, in [10] a heuristic approach was adopted, while in [13] an approach based on the expectation maximization (EM) was proposed.

Although down-sampling by excluding is straightforward to implement in practice, it has a critical drawback of information loss and may lead to inaccurate models, mainly if the hard-to-measure variable is sampled scarcely and/or with uncertain delays [13]. A better approach is to model the data by using the FIR filter. However, the weights and length of the FIR filter should be designed or estimated carefully.

2.2. Missing Data

It is quite common to have observations with missing values for one or more variables. The problem of missing data occurs when no value is stored for a variable in an observation. There are two common approaches to deal with missing data. The first one is the removal of samples containing missing data, an approach also known as listwise deletion. The second approach is to fill-in the missing values using some imputing method. The first approach can be used if the number of missing values is small, but otherwise it should be avoided [7]. In the second case, the simplest strategy is to impute the missing value with a mean or median of non missing values for that variable. Another approach is the hot-deck imputation, where a missing value is imputed from a randomly selected value of the input for similar target values [14]. These methods of mean/median imputation, and hot-deck imputation, are usually referred as multiple imputation.

Two other methods which are often employed for handling missing data are the maximum likelihood (ML) method and the EM method. The ML method models the missing variable/s based on the available data. Essentially, the ML assumes some model for the data distribution of the missing variable, and then the parameters of the model are estimated using ML. In [15] the authors assumed linear relationships, while in [16] several nonlinear models were used to model the relationship among the non-missing variables and the variable with missing values. In both cases, the authors reported significant improvement when compared to multiple imputation methods (hot-deck, and mean/median imputations). The EM approach to handle missing data is reported in [17], it works similarly to the ML procedure, although it is an iterative procedure. First it estimates the missing data using the observed data and the first estimates of the model parameters. In the second step, the estimated missing data are used together with observed data to estimate the parameters. This iterative process repeats until there are no significant changes in parameters estimates. In [18] it is made an extensive review on methods for missing data imputation.

2.3. Outliers

Outliers are observation values that deviate significantly from the typical, meaningful range of values. Observations take inconsistent values when compared to the majority of recorded data, and this can greatly affect the performance of the SSR design [2]. Outliers can be caused, for example, by sensor malfunction, communication errors, or sensor degradation. To alleviate the effects of outliers it is necessary first to detect them, and then to treat them.

However, when applying outlier detection methods, usually the results have to be validated manually by the model developer and/or process expert. The goal of the manual inspection is to detect any possible outlier maskings (i.e. false negative detections - not detected outliers) and outlier swamping (i.e. false positive detections - correct values labeled as outliers).

Typical outlier detection methods are based on statistical techniques. The most simple approach is the 3σ-rule [19], which is based on an univariate distribution of variables. The 3σ-rule works as follows: assuming that a variable is drawn from a Gaussian distribution with mean μ and standard deviation σ, the samples of that variable which are outside the bounds $[\mu - 3\sigma, \mu + 3\sigma]$ are considered outliers. A robust version of 3σ-rule is the Hampfel identifier [20], which considers the absolute mean and absolute mean deviation. The Hampfel identifier is suitable in the cases where the data is severely affected by outliers, and it has shown to be practically effective in real applications [21; 19]. The above approaches are considered as univariate outlier detection methods, since they are applied on each variable separately. However, in many cases outliers cannot be detected by considering the variables individually. Then, multivariate techniques should be adopted. Outlier detection based on multivariate techniques takes into consideration the interaction among variables, and it can deliver most accurate results, as demonstrated by [1; 22]. It often works by using distance measures to indicate those samples which are far from the center of data distribution. A common distance measure adopted is the Mahalanobis distance, where the samples considered outliers are the ones with a large value of Mahalanobis distance [23]. Other multivariate approach commonly used in the SSRs context is based on data projection/dimensionality reduction techniques, such as principal component analysis (PCA) or partial least squares (PLS), together with the Jolliffe parameters [24; 25]. It works by decomposing the original data using PCA or PLS, and then using the decomposed data to compute the Jolliffe parameters [24]. The Jolliffe parameters help to identify the samples that do not conform with the correlation structure of data and the ones that inflate the data variance. In [25; 1] outlier detection based on PCA, PLS, and Jolliffe parameters was studied and has been shown to be a powerful alternative for outlier detection in SSRs applications.

In [26] several outlier detection methods were compared (six in total), and the authors concluded that the efficacy of the proposed methods depends strongly on the problem domain. In particular, the efficacy depends on whether the data is multivariate normal, on the dimension of data set, on the type of outliers, and on the amount of outliers in the data set. The authors recommend a battery of multivariate outlier detection tests to detect outliers. In the SSR context, [22] compared several outlier detection methods in the modeling of a sulfur recovery unit. The use of outlier detection improved considerably the SSR accuracy in the case-study, and PCA-based outlier detection achieved the best results.

The book of [27] provides several discussions regarding pre-processing techniques and their application in the SSR context. Real-world examples as well comparison of techniques are also presented. In [2; 28] general overviews on pre-processing techniques are also presented.

3. Variable Selection

In SSR applications there is frequently a large amount of candidates for input variables coming from the supervision structure of the process. The number of candidates can range to thousands [29; 30]. The use of black-box models already suggests that the SSR designer has few knowledge about the system to be modeled, and consequently about the variables which affect the target variable. However, this not true in all the cases, since in most of SSRs applications the selection of a set of most relevant variables is made by system experts. Nonetheless, for physically large and highly integrated processes, enumeration and selection of candidate variables based on process insight may not be feasible [25]. Moreover, most of the works in the literature indicate that frequently only few variables are necessary to compose the SSR model. A reduced number of variables has several advantages, such as the reduction of model development time, possibility of aggregation of the information about the physical interpretation of the process, or the improvement of the model performance. Moreover, a reduction of the number of variables implies a lower number of required real sensors, decreasing costs, and increasing or enabling feasibility of applications.

The following are possible approaches concerning variable selection that may be adopted during SSR design [31]:

Use of all inputs: This approach leads to extremely high dimensional approximation problems. The problems associated with learning of a model with many input variables suffer from large computational demand, large probability of occurring overfitting, and poor performance of the regression model. Overfitting means that the model is very accurate on training data, but it has poor accuracy on previously unseen test data. A large number of input variables and a limited number of samples causes a curse of dimensionality phenomena [32], which refers to some, normally problematic, phenomenon that occurs in high-dimensional spaces but does not occur in low-dimensional spaces. In the case of a variable selection setting, one curse of dimensionality problem that occurs is that the number of samples required to represent an input space increases exponentially with the number of variables. Another problem that occurs is the increase of computational costs in algorithms dealing with high-dimensional spaces. Variable selection is one way to prevent overfitting, increase the model performance, and also to avoid the curse of dimensionality phenomena;

Unsupervised variable selection: The typical approach for unsupervised variable selection is based on principal component analysis (PCA) [24]. It works by projecting the input space into a latent space, where the first latent variable (also called principal component) has the largest possible variance (i.e. it accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e. uncorrelated with) the preceding components. Then, few components obtained by PCA are used to learn the model. The selection of the number of latent variables is crucial to attain satisfactory results. In a recent paper [33] discusses the ways to select the number of components to retain in a PCA. Applications of PCA as a basis for unsupervising variable selection are vast in SSRs literature [34; 35; 36];

Supervised variable selection: In this approach the selection of input variables is directly guided by the goal of attaining the highest possible model accuracy; the relation between the model accuracy and a subset of inputs can be accessed independently or dependently of the model. Any procedure for input variable selection must be based on two main components [37]. First, a criterion to measure the quality of a subset must be defined, to judge whether one subset is better than another (this is usually refereed as cost/fitness function). Second, a search procedure must be defined to search through candidate subsets of variables. The selection criteria can be classified into three different classes: filter methods, wrapper methods, and embedded methods [38; 39]. Filter methods use statistical measures (e.g. correlation coefficient (CC), mutual information (MI)) to quantify the quality of a subset, and are independent of the model used. On the other hand, wrapper criteria use the performance of the model as the criterion, using for example the mean square error (MSE), the Akaike information criterion (AIC), or the Cp statistics (all these methods will be later explained in Section 4). In the third class, the embedded methods use a specific caracteristic about the model itself or the process of model learning to define the criterion (e.g. pruning methods, regularization). For all the three classes of methods, to achieve the optimal solution, the search procedure can consist of an exhaustive search of all possible subsets of variables. However, exhaustive search is highly computationally/time expensive, even for a moderate number of input variables. Then, in practical applications, simplified search methods such as sequential search, or stochastic search are usually employed in order to limit the computational complexity of the search procedure. Appendix A gives an overview on search procedures.

3.1. Filter Variable Selection

The use CC is the most popular method employed for input variable selection in SSRs. In such CC variable selection method, the linear strength between each input and the target is computed using the Pearson correlation coefficient, and the variables are ranked according to their strength [1; 40; 41]. For nonlinear regression settings, the Pearson correlation is usually replaced by the univariate mutual information (MI) [42], and similarly to CC-based methods the variables are ranked according with their importance (see ranking search in Appendix A). The variable ranking algorithms based on the correlation coefficient and/or univariate MI can be used as the principal selection mechanism or as an auxiliary selection mechanism [39]. As a principal selection mechanism, the selected inputs are used in the learning of the regression model. As an auxiliary mechanism, the variable ranking is used as a kind of screening step, removing only irrelevant variables, and then the remaining variables are passed to another variable selection algorithm to finally select the variables.

The multivariate MI approach for variable selection is a extension of the univariate MI approach, and it measures the dependency of a set of input variables on the target. In [43] it was demonstrated that the multivariate MI is an adequate criterion for variable selection in regression settings. However, the estimation of multidimensional probability density functions (pdfs) in the multivariate MI approach is not an easy task: sparcity of data, and high computational demand are some problems associated with this task.

In SSRs/regression applications, the nonparametric k-nearest neighbors algorithm (KNN) [44] and the histogram based estimators are the most commonly employed methods for pdf estimation in the multivariate MI approach [45; 46]. The KNN approach tends to be used because of the good results reported in the literature [47; 48], and the histogram method is used because of its easy implementation and good results when working with a small number of variables [49].

However, when dealing with a large number of input variables, the use of multivariate MI as a quality criterion for evaluating subsets of variables is not adequate. The problems associated with pdf estimation are highly aggravated with the increase in problem dimensionality. In [50], instead of estimating the multivariate MI, the authors approximate it by using the univariate MI. In the work of [51], inspired in the work of [50], the authors developed an algorithm called as the “minimum redundancy maximum relevance” (mRMR) principle for variable selection based on univariate MI. It is a well accepted method for variable selection (with more than 3313[1] citations since 2005). Furthermore, in [52] it was demonstrated that the algorithms

[1] According to Google Scholar

of [50; 51] are equivalent to maximization of the multivariate MI between inputs and the target. Another variant of [50; 51] was proposed in [53] and is called normalized mutual information feature selection (NMIFS). The NMIFS criterion changes the form of how the mRMR criterion is defined, to reduce its bias and improve the quality of the selection of variables.

Several applications of MI in SSRs and related areas have been developed. In [49] a combination of genetic algorithms (GAs) and the mRMR principle was used to select the dynamics (i.e. time lags) of input variables of a MLP model. In [54; 29], the discrete mutual information was used to select the variables and corresponding time-lags in different SSRs and regression problems. In [29], it has been demonstrated that the KNN estimator of multivariate MI, together with the sequential forward search (SFS) procedure (see Appendix A), has a superior performance when compared with the CC variable selection method in two SSRs problems. In [29], the selected variables were employed in a support vector regression (SVR) model to predict the targets. In [55; 56], the KNN estimator of multivariate MI, together with the SFS procedure was successfully employed as a variable selection tool in several real-world case-studies, and the model utilized was the MLP model. Another recent filter method for input variable selection was based on the nearest correlation spectral clustering [57]. The PLS model was learned with the selected inputs and then used for estimating the ethane concentration in an ethylene fractionator.

3.2. Wrapper Variable Selection

Another approach for selecting input variables is by assessing the performance of the learning model (wrapper approach). Usually this approach achieves more accurate prediction results when compared with filter methods, because it takes into account the approximation model. However, in the wrapper approach it is necessary to learn a regression model every time a subset of variables is going to be evaluated, which is therefore computationally expensive. Applications of wrapper methods in SSRs/regression applications are given below.

In [58], to overcome the problem associated with a limited number of samples and a large number of inputs, a bootstrapping resampling on data was applied. Then, a sequential forward float search (SFFS) (an improved version of SFS; see sequential search in Appendix A for an explanation on the SFFS procedure) together with a linear model (LM) with its parameters estimated by the least squares (LS) estimator, was used to select the relevant variables. The error of the LS model was used as the cost function. The selected variables were used in a PLS method to predict the vinyl chloride in a polimerization process. The reason for the use of LS instead of PLS, in selecting the variables, lies in the fact that LM has low computational cost when compared to PLS model.

A genetic algorithm (GA) (see stochastic search in Appendix A) together with the PLS model was applied in [59] to select the input variables. Another method based on GA and PLS to select the variables and the dynamics of the system (i.e. the time lags) was proposed in [60]. In both these two works, the error of the PLS model was used as cost function.

In [61] a vision-based model was developed for the prediction of ore quality at the mine level. Due to the large number of available variables, a GA combined with a MLP network was applied to select the most relevant variables. The MLP error was used as the cost function.

To select the variables and the dynamics of the system, a SVR model together with a variant of GA encoding [62] was used in [63]. The SVR error was used as the cost function. In [64] the variables and the parameters of a SVR model were determined using a hybrid genetic simulated annealing search. To select the models with a complexity as small as possible, the fitness function was based on the AIC.

In [65] the input variables were selected based on their individual prediction performance, based on the error of a Takagi Sugeno (TS)-fuzzy model. The authors compared selection performed by the expert with the automatic selection of the inputs, and it was concluded that both approaches are competitive, but in the presented case of study, better results were achieved with the automatic method.

In [66] variable selection based on MLP model and sequential backward search (SBS) (see sequential search in Appendix A) was studied. Discussion about the stopping criterion, accuracy, and computational time was performed. The authors concluded that the MLP together with SBS provides good results, but the main problem regarding this approach is its demanding computational time.

3.3. Embedded Variable Selection

Embedded algorithms form a class of variable selection algorithms where the selection of variables is embedded within the model or the model learning. They share similar characteristics with the wrapper algorithms, so it may be difficult or confusing to distinguish between embedded and wrapper approaches in some cases [67]. However, the main difference between them is that an embedded method which is based on a specific model cannot be used/employed in combination/integration with another model.

Regularization methods are a class of embedded variable selection approaches. Such methods work by adding a penalty term to the model parameters in the model error function. This penalization shrinks the freedom of the model parameters during learning. For linear models they are used as an alternative to the LS solution, and in cases of poorly conditioned or ill-conditioned problems. From the statistical theory, the most well know regularization methods are the least absolute shrinkage and selection operator (LASSO) [7], ridge regression (RR) [68], and elastic net (EN) [69]. Another regularization method, widely employed in the chemometrics theory, is the PLS. In [70] the authors give the statistical point of view on the PLS, and concluded that PLS plays a role similar to the RR.

The regularization approach can also be expanded to application in neural networks (NN), by adding a penalty function in the error function. A penalization method which penalizes both useless input variables and hidden nodes was proposed by [71]. It was shown that the method outperforms the traditional regularization methods for weight decay penalization [37] and input decay [72].

In predictions settings based on NN models, variable selection can be based on sensitivity analysis approaches, also referred as pruning methods [73; 74]. In sensitivity analysis, the importance of an input is measured by computing the variation of the output when the input is perturbed. Usually, all inputs are used to train the network, and then irrelevant inputs are removed sequentially if they are considered irrelevant from the sensitivity metric point of view. After the removal of irrelevant variables, the model is retrained and the sensitivity analysis can be performed again. This procedure continues until the results get satisfactory. This is the same procedure as the SBS search (see Appendix A). Garson [75] proposed a metric of importance based on the weights of the NN input layer. Several other proposed methods evaluate the relevance of a certain variable by computing the partial derivatives of the output with respect to that variable [76; 77]. In [78] the importance is measured by varying the values of one variable while keeping all the others untouched, and the input variable whose changes mostly affect the output is the one that has the most relative influence. In [79] a NN is trained with all variables, and then useless variables are sequentially removed according to an exclusion criterion based on the sensitivity metric proposed in [75]. However, in contrast with [75], when a variable is removed the existing NN model is adjusted with a lower computational cost when compared to performing again a complete retraining of the network.

A majority of the embedded methods proposed for support vector machine (SVM) models are targeted for classification tasks, but some methods can be easily extended from classification to regression [80]. Despite their applicability, their use on SSR applications has not been tested yet, but they are worth mentioning here. Input selection based on SVM models proceeds in the same way as in MLP input selection based on sensitivity analysis, i.e. the selection process is usually performed as follows: train a SVM with all variables, select and remove the least relevant variables according to the sensitivity metric, re-train the SVM model and proceed in the same manner until satisfactory results are obtained. In [81] the input weights of the SVM model were used as the sensitivity metric. The approach was applied in a cancer classification problem where the number of inputs is larger than 7000 and only few samples were available. A different approach to define the sensitivity metric was adopted by [82], where the sensitivity metric was based on the upper bound of the leave one out cross validation (LOOCV) error of the SVM model.

The embedded variable selection method based on the SVR model which is proposed in [80] is primarily devoted to regression. It exploits the characteristic that the SVR output can be interpreted as the conditional density function of the target, given the input variables, under the assumption that the output error is characterized by a Laplace or a Gaussian probability distribution (such interpretation that the output error is characterized by the Laplace or the Gaussian probability distributions is demonstrated in [83]). Thus, the proposed sensitivity metric measures the difference over the input variable space of the conditional density functions of the SVR prediction with and without the feature.

3.4. Hybrid Approaches

Several SSRs applications combine several methods to promote the selection of input variables.

In [84; 1] a combination of three variable selection methods was used to select the variables. The methods used were the correlation coefficient/scatter plots, partial correlation, and the Mallows Cp statistics [85]. The scatter plots and correlation coefficient were used as pre-filtering, to form a preliminary subset. Then, the Cp statistics and the partial correlation were used to aid in the selection of the best subset.

In [25], PCA pre-processing was applied on the variables as an unsupervised variable selection. It provided better results when compared with the variable selection methodology used in [84; 1] (discussed in the previous paragraph). In [86], it is demonstrated that collinearity increases the variance of the MLP model, and then it is proposed to use the PLS as a pre-processing step for a MLP model, since PLS eliminates the collinearity in the input space. The PLS together with a MLP model provided good results when compared to a single MLP.

In [40] the input variables of a fuzzy model are preselected from the variables of the dynamical process by means of correlation coefficients, Kohonen maps and Lipschitz quotients.

In [87] a hybrid approach based on wrapper and embedded methods was proposed. It approximates the response/results of variable selection based on the MLP prediction error and the SBS search procedure, defined here as SBS-MLP, but with much less computational effort. The proposed method presents similar or better approximation performance when compared to two filter methods based on MI criterion proposed in [51] and [53], the embedded method proposed in [79], and the wrapper method based on SBS-MLP [88; 66]. Moreover, it has been shown that the proposed method has similar prediction performance when compared to the traditional SBS-MLP algorithm, and has the advantage of having lower computation cost. The proposed method presents similar or better approximation performance when compared to the other four methods.

4. Model Choice and Training

There are two distinct model approaches applied for SSRs development. The first is based on white-box models, obtained through a physical knowledge of the process, and the second class is based on black-box or data-driven models, based exclusively in constructing a model from empirical data of the process. Modeling by the white-box approach requires strong knowledge about the process and demands a long time of modeling work to build the models [89]. It usually focuses on the description of the ideal steady-states, not being able to describe the real process conditions [2]. For complex systems, the white-box modeling approach may be virtually infeasible. Black-box or data-driven models are based on empirical observations of the process (the methods themselves are empirical predictive methods). Black-box modeling is able to describe real conditions of the process, and it requires few knowledge about the system to be modeled. Nevertheless, it requires intensive work on process data. Some difficulties with these types of approaches are related to the difficulty of choosing the correct model type and structure, the functions to be used, and the quantity of function terms necessary for the development.

In black-box modeling, the first aspect to decide about is which kind of model is going to be used. There are always two choices: a linear model or a non-linear model. According to many authors, a linear model should always be considered before a nonlinear model. If the linear model does not provide satisfactory results, one possible explanation, besides many other possibilities, is that the system possesses a non-linear behavior, then a non-linear model should be the best choice [31]. Good overviews of black-box structures for regression ranging from linear models (e.g. PLS, LASSO, RR), to nonlinear models (e.g. NN, SVR, Fuzzy Systems (FS)) are reported in the classical books [6; 5; 31; 7; 4].

The most popular data-driven models used in SSRs applications are the linear models with LS or PLS estimation methods [90; 91], PCA [24] in combination with a prediction model, NNs (mainly the MLP structure), SVRs, FS, and Neuro-Fuzzy Systems (NFS) [92; 93; 94]. The PLS solution is the preferred and mostly applied solution in combination with linear models when comparing to LS, since it can handle data-collinearity, which is a common characteristic in industrial applications.

Soft sensors are not always composed of a single regression model. A combination of a collection of models is often employed. This is denominated an ensemble approach, which forms an ensemble of models. Ensemble methods play an important role in SSRs applications, mainly when the number of samples for modeling is small [95]. The ensemble of NN models was detailed and discussed in [96], where the authors proposed a method for building an ensemble of NN models based on GA. A related approach was used in [97] where a framework to optimize the structure of an ensemble of MLP models was presented. Several MLP models with different structures were trained using the bootstrap resampling. Then, GA and simulated annealing (SA) were used to perform the optimization of the model architecture. In [98], an evolutionary ensemble learning using NN and based on negative correlation learning was presented. However, [98] has some shortcomings such as not considering the possibility of linear combination among models, and using pre-defined models' architectures.

Fuzzy models are knowledge-based models. In some complex applications it is difficult to tune such models. Some approaches try to overcome this difficulty by optimizing the fuzzy model using evolutionary algorithms. In [40] the TS-fuzzy model is tuned using a GA-based approach. In [93] the work of [40] was expanded to learn the TS-fuzzy TS structure together with the selection of input variables and delays.

In almost all soft sensor applications, a single model is tuned using all available training samples, without distinguishing the operating modes of the process. However, the existence of multiple operating modes in a process is an inherent characteristic of most industrial applications. Sometimes, multiple operating modes result from external disturbances, as for example a change in feedstock or product grade or even changes such as the diurnal load variation of a power plant or the summer-winter operation of a refinery [99; 100]. In these situations, consistently training a model for each operating mode or for each set of correlated operating modes of the process has been shown to be reasonably consistent and to be beneficial for the prediction accuracy [101; 102]. During online operation, when a new sample is made available, the model which is the most adequate for the new sample is identified and then used to make the prediction. The identification of which model will be used is a key issue in the development [102; 103; 104], which can be done using expert knowledge [102] or using automatic tools, such as finite mixture of Gaussian models (FMGM) [101].

In this context, in [102], the authors work on modeling the operating modes in a polymerization batch process case study. The correlated operating modes have been grouped, and then a separate PLS model was tuned for each set of correlated operating modes. During online operation, the incoming sample is assigned to the corresponding mode and its model is used for the prediction. However, in [102], the expert knowledge of the operators was used to determine the operating modes and in some cases or problems such information might not be available.

Another approach, based on the FMGM, was proposed in [101]. In this work, the FMGM is used to automatically identify the different operating modes of the process. Then, multiple localized Gaussian process regression models in the nonlinear kernel space were built to characterize the different dynamic relationships between process variables (inputs to the prediction setting) and quality variables (outputs of the prediction setting) within the identified operating modes. During online operation, the incoming sample is assigned automatically to the corresponding submodel, using the FMGM. The major drawback of [101] is that the determination of the operating modes and model tuning is done separately, i.e. the set of operating modes is determined independently of the model used. However, as verified in the case study of [102], a model can be set for more than one operating mode, with the advantage of reducing the number of necessary models and increasing the number of samples available for tuning each model. Another drawback of [101] is that the number of samples used for tuning each model is constrained by the number of samples of each operating mode, which is defined by the FMGM. The approach of [101] leads to "hard“ partition boundaries, and consequently just a part of the total of samples is used for tuning the prediction model of each operating mode. Such an approach can lead to poor modeling on the corresponding operating mode, depending on the chosen model and the available samples

In [105] a method for dealing with online prediction of the quality variables in processes with multiple operating modes is proposed and derived. The method is called mixture of partial least squares (PLS) experts (Mix-PLS). The Mix-PLS was be derived based on the mixture of experts (ME) framework [106] and the PLS algorithm. The ME models input-output observations by assuming that they have been produced by a set of different random sources (the random sources can be thought of as operating modes) and the parameters of each expert, and of each gating function, were determined using the PLS algorithm. It was demonstrated that the solution of the parameters using the PLS algorithm overcomes the problem of collinearity of input data and also makes the Mix-PLS less prone to overfitting with respect to the number of mixture models. The Mix-PLS was compared with the SVR, MLP, Linear and PLS models, with superior performance in all the presented cases of study.

5. Model Validation

The objective of the model validation step is to evaluate the capability/ability of the trained model to perform generalization to new samples. Generalization accuracy can also be used as an estimator for model ranking in a variable selection approach (e.g. in wrapper variable selection) [9]. For a large data set, usually the model is learned using only a part of the data set and then the model performance is measured on the remaining data, usually called validation data set, using some performance metric, usually the MSE (e.g. lower values of MSE indicate better models) or the normalized root mean square error (NRMSE). The NRMSE is a normalized version of MSE, often expressed in percentage, which gives a more intuitive analysis on the performance of the model. For small data sets, a cross-validation technique is usually employed to evaluate the performance of the model. The common cross validation techniques are the K-fold cross validation and the leave-one-out cross validation (LOOCV). In K-fold cross validation, the training data set is randomly split into K folds, and then the model learning is performed using the samples from $(K-1)$ folds, and the resulting model is evaluated on the remaining fold, using some performance metric. This process is repeated for all K folds, and the performance of the model is the average of the performance metric on the K folds. The LOOCV is usually employed when the number of samples is very small, and it is equivalent to the K-fold cross validation when the number of folds K is equal to the number of samples. Other approaches measure the quality of a model in terms of its accuracy-complexity trade-off (ACT), using criteria such as the AIC [107], the Bayesian Information Criterion (BIC) [108], or the Cp statistics [85].

For dynamic linear systems, the autocorrelation function of the residuals and the cross-correlation functions between the residuals and the input over a set of unseen data [109] are usually employed to evaluate the capability of the trained linear dynamic model. For non-linear dynamic systems, the work of [110] has provided several metrics to evaluate non-linear dynamic models based on NN.

6. Soft Sensor Maintenance

During SSR design the historical data of the process is used to learn the SSR model. However, the historical data contains limited information, corresponding to a limited period of time, and possibly also focusing on a limited set of operation areas of the state space. When dealing with new events, not described in the historical data, the SSR tends to decrease its performance. In this context, and to overcome such performance deterioration, the objective of SSR maintenance is to maintain a good SSR response even in the presence of process variations, or some data change. Generally, this is done by updating the SSR model online/recursively, in batch or sample wise mode, using the incoming samples of the process (in this context the SSRs are called "adaptive SSRs" [111]). From the machine learning perspective, the area of adaptive SSRs is related to the problem of concept drift. Concept drift means that the statistical properties of the target variable changes over the time, the term concept means the object/target to be predicted [112].

There are three types approaches commonly employed in dealing with concept drift: (1) sample selection, (2) sample weighting, and (3) ensemble learning (or learning with multiple concept descriptors) [113]. Moreover, as already discussed before, the mostly used models in SSR applications are based on multivariate statistical methods (LS, PLS, PCA) or artificial intelligence techniques (NNs (mainly the MLP structure), SVRs, FS, and NFS). In adaptive SSRs such models can also be employed, but there is the concern regarding the learning/adaptation of parameters. The model(s) can be applied as a single model, in the sample weighting or sample selection approaches, or several models can be applied together in the ensemble approach.

6.1. Sample Selection

In sample selection, the idea is to select relevant samples related to the current concept. The next step is to use such samples to update or retrain the existing model. Normally, this selection is done using window-based approaches, where the samples which are inside of a window are used to retrain/update the model, while samples outside of the window are discarded. The issues of selecting the size of the window and deciding when to retrain/update the model are crucial for a successful implementation. If the selection of the window size is poorly handled, there is a danger that the SSR adapts to noise (if the window size is too short) or, in the case of a too long window, it can lead to limited adaptation capability [114]. Some adaptive methods based on ANN models in the sample selection strategy were proposed in the literature. In [115; 116], a moving window was adopted to retrain the ANN model. When a new batch of samples is available the old data is dropped out of the window and the neural model is retrained adapting to the concept of the new data. In [116] the most relevant features were selected offline using the first part of the training data by using a forward search procedure in combination with a MLP network.

Adaptive learning methods for NFS and SVR have been proposed in the literature, and they are usually based on sample selection or ensemble learning. NFS are widely applied for prediction [2; 93], but their parameters are usually learned offline. Online tuning of NFS can be done by Evolving Fuzzy Systems (EFS) [117]. A step-wise online learning algorithm for SVR training was proposed by [118], where the update can be done by removing or adding new support vectors, an application in the soft sensor context is given in [119]. In [120] it is proposed the Adaptive Kernel Learning (AKL) framework for prediction and monitoring tasks. In this case, the SVR optimization problem was solved by the least squares approach [121]. In [122], an adaptive kernel learning method was used. The examples were selected, and the exclusion of redundant examples was performed to reduce the complexity of training. It was shown to be superior to RPLS in the presented case of study.

6.2. Sample Weighting

In the sample weighting strategy, the samples are weighted according to their age (the importance of the samples decreases over time). The learning/adaptation of parameters is usually done using adaptive learning by means of exponentially recursive learning. The adaptive learning has relation to the recursive or online learning where each sample is presented once and only once to learn/adapt the parameters, but in adaptive learning there is the ability to forget old examples by exponentially assigning low weights to old samples, usually by setting a forgetting factor $0 < \lambda < 1$, such that the model could capture the information of the recent data [113; 111]. Using such sample weighting approaches, there is no need to use memory to store the samples.

In the sample weighting approach, the following learning strategies have been used in the literature for the LS, PLS, and artificial neural networks (ANN) models. For the LS solution, there is the recursive LS (RLS) method, which is a well known example of recursive learning, where the coefficients of a linear model that minimize the linear least squares cost function are recursively computed. The PLS is implemented with its recursive/adaptive form, the recursive PLS (RPLS) [123]. It is the most popular method in adaptive SSRs [124; 125; 126; 127; 128; 129; 59; 11; 130]. For the other state of the art methods, there are some adaptive learning strategies in the literature. For single layer feedforward ANN, a fast learning algorithm with offline and online solutions, called online sequential extreme learning machine (OS-ELM) was proposed in [131]. All these methods are able to forget old samples by setting a forgetting factor. In [132] the problem related with exponential weighting of samples in adaptive soft sensors was studied. It was assumed that when learning the adaptive models with small values of forgetting factor, the model suffers from problems similar to the ones associated with learning of static models with small number of samples. Then, based on this, a mixture of low dimensional models was proposed and derived, based on the mixture of univariate linear regression models. Mixtures of other types of models, possibly nonlinear, but linear in the parameters were also considered. The proposed method was evaluated in two time-varying real-world data sets, and compared in different settings with the state of the art methods in adaptive soft sensors. The proposed method demonstrated to provide the best results in almost all cases, mainly when using small values of forgetting factor.

6.3. Ensemble Learning

In the ensemble learning strategies, the goal is to construct a model for each concept in the data distribution. When a new input arrives, the final prediction value is a combination of the results of all the models built previously for all the concepts, such as a weighted average of such results. Moreover, in the ensemble method, there are two possible areas that may be subject to adaptation: at the level of the model combination, or at the level of the models. The ensemble method is less attractive because of its computational demand, necessary to process and store several models and/or samples.

Ensemble learning methods find different concepts in the historical data and learn a model for each of these concepts. In [11] a PLS model was constructed for each different concept found (an approach based on the PLS model error was used to determine the different concepts).

The final prediction is a combination of the set of the available models, where the combination takes into account a probability of each model being responsible for the data to be predicted. The adaptation is performed at the level of model combination and at the level of recursive adaptation of the models. The authors termed this method the incremental local learning soft sensing algorithm (ILLSA). [133] developed a SSR method using an ensemble learning strategy where a clustering method, based on the fuzzy C-means clustering (FCM) algorithm, was used to find different concepts, and then a SVR model was learned to predict in each concept. During online operation, when a new sample arrives, the FCM algorithm sets the corresponding adequate SVR model to be used to predict the output.

7. Conclusions

The soft sensor technology has important potential for industrial applications and academic research. From the industry perspective, the soft sensor has an enormous potential to be used as a commercial tool to improve performance, efficiency, automation degree, and output quality in industrial systems. From the academy/research perspective, the soft sensors can be stated as a multidisciplinary topic of research, that encompasses several areas of study, such as machine learning, pattern recognition, artificial intelligence, system identification, and statistical learning theory. Moreover, it has several topics to be researched, where the most emergent topics, are the problem of variable selection (including dynamic selection) and soft sensor maintenance. Another topic of research, is regarding the learning of soft sensor models in multiple operating scenarios/modes.

Acknowledgments

The authors acknowledge the support of Project SCIAD "Self-Learning Industrial Control Systems Through Process Data" (reference: SCIAD/2011/21531) co-financed by QREN, in the framework of the "Mais Centro - Regional Operational Program of the Centro", and by the European Union through the European Regional Development Fund (ERDF). Francisco A. A. Souza and Jérôme Mendes have been supported by Fundação para a Ciência e a Tecnologia (FCT) under grants SFRH/BPD/99708/2014, and SFRH/BPD/112774/2015.

Appendix A. Search Procedures

In a variable selection algorithm, a search procedure is used to guide the search for the best subset of variables. For D input variables, there are a total of $2^D - 1$ possible subsets, where only some of the subsets attain the optimal solution. Typically, the optimal solution may be attained for only one of the subsets. By searching over all possible subsets (this is called as exhaustive search), it is possible to lead to the optimal solution. However, for exhaustive search there is the problem of the large computational demand. For example if there are only 20 variables, i.e. $D = 20$, there are 1048575 solutions that need to be evaluated, if the criterion to evaluate one subset takes approximately ~ 1(sec) (being optimistic), then it would be necessary ~ 12 days to select the best subset. The branch and bound (B&B) algorithm leads to the optimal solution with less complexity than the exhaustive search, under the constrain that the evaluation function must be monotic [134]. However, the algorithm still has an exponential worst case complexity, which may render the approach infeasible when a large number of candidate variables is available [39].

The large computational costs associated with the exhaustive search and B&B algorithms, caused by the necessity to evaluate so many subsets, can be reduced by using search strategies that prioritize the computational time rather than the quality of the solution, while still providing good results. Such strategies are based on rankers, sequential and stochastic searches. These techniques are briefly reviewed below.

Appendix B. Ranking Search

The ranking search proceeds as follows. First, the importance of each input variable, with respect to the target (measured by any criterion, e.g. CC, MI), is computed. Then, the variables are ranked according to their individual merit, with respect to the target variable, in accordance with the chosen criterion. Then, only a subset of the top variables (from the ranked set), are selected, and the remaining variables are excluded. In this search approach only D evaluations are required; a very fast approach. This method gains on the speed of selection, but loses on the quality of the selected variables. This happens because, the variables are selected without taking into consideration the interaction among them.

Appendix C. Sequential Search

The sequential search works by removing or adding variables sequentially, following a certain order. The most common sequential search procedures are the sequential forward selection (SFS) and the sequential backward selection (SBS). The SBS procedure, proposed by [88], starts with all variables, and at each step the variable that contributes least to predict the target, according with the subset evaluation criterion, is removed. The SBS procedure stops when a pre-specified number of variables are removed or until the results get satisfactory. The SFS, introduced by [135], starts with an empty subset, and at each step the variable that mostly contributes to predict the target, according with the subset evaluation criterion, is added to the set of selected variables. These methods are largely used in variable selection procedures.

Both SFS and SBS have the same complexity in the worst case scenario (it is necessary to evaluate $\frac{D(D+1)}{2}$ subsets), but in a practical perspective the SFS executes faster than SBS. This happens because the SFS algorithm evaluates smaller subsets than the SBS at the beginning of the search.

The major problem related to the SFS and SBS approaches is that, for example, when a variable is removed in SBS, it cannot be selected again. This results in the so called *nesting effect*, i.e. bad decisions made at the beginning of the search cannot be corrected later. To avoid or alleviate the nesting effect in the sequential selection Stearns [136] proposed the Plus-l-Minus-r search method. Each iteration of the Plus-l-Minus-r is divided into two substeps. In the first step, the SFS runs to select l new variables, and in the second step the SBS runs to exclude r variables from those that have already been selected. Pudil [137] proposed modifications on the SFS and SBS to allow them to reselect removed variables, then avoiding the nesting effect, they are called as sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS), and their idea is similar to the Plus-l-Minus-r algorithm.

Appendix D. Stochastic Search

Stochastic methods are optimization methods which include some randomness in the search procedure. This can be thought as a good strategy when dealing with a large number of input variables [138], since it corresponds to search randomly over the input space, but following a certain heuristic. The class of stochastic algorithms includes, but is not restricted to, Genetic Algorithms (GA), Ant Colony Optimization (ACO), and Simulated Annealing (SA).

The GA is inspired by the biological evolution, more specifically by the Darwinian principles of natural evolution, where the best individuals have a high probability of survival; It was first introduced in [139]. In the GA, solutions are encoded into chromosomes (individuals) and the fittest ones are more susceptible, have higher probability, to be selected for reproduction, producing offspring with characteristics of both parents. For some of the offsprings an operation called mutation (inspired by the natural evolution) is applied, to include diversity in the solution.

The ACO is an optimization methodology based on ant behaviors to establish the shortest route paths from their colony to food sources and back [140]. In nature, ants randomly walk for finding food, then they return to their colony while laying down pheromone trails. Other ants, when finding such path, tend to follow the trail and when they find food, they also walk back to the colony laying down pheromone, thus reinforcing the trail.

SA is a meta-heuristic proposed in [141] for global optimization problems. SA is inspired in the behavior of a warm particle in a potential field. Generally, a particle tends to move down, to the lower potential energy, but since it has kinect energy (caused by the non-zero temperature), it moves around with some randomness, and occasionally it jumps to higher potentials. The particle is annealed when the time passes in this process, i.e. if temperature decreases gradually, so that the probability to move upwards decreases with time. In SA, the solution is represented by the particle and the potential energy represents the cost function.

[1] L. Fortuna, S. Graziani, A. Rizzo, M. G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes, 1st Edition, Advances in Industrial Control, Springer, 2006.

[2] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Computers & Chemical Engineering 33 (4) (2009) 795–814.

[3] S. Wold, Chemometrics; what do we mean with it, and what do we want from it?, Chemometrics and Intelligent Laboratory Systems 30 (1995) 109–115.

[4] C. M. Bishop, Pattern Recognition and Machine Learning, 1st Edition, Springer, 2006.

[5] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999.

[6] L. Ljung, System Identification: Theory for the User, 2nd Edition, Prentice Hall, 1999.

[7] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York, 2001.

[8] L. Ljung, Perspectives on system identification, Annual Reviews in Control 34 (1) (2010) 1–12.

[9] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd Edition, Wiley-Interscience, 2000.

[10] Y. Wu, X. Luo, A novel calibration approach of soft sensor based on multirate data fusion technology, Journal of Process Control 20 (10) (2010) 1252–1260.

[11] P. Kadlec, B. Gabrys, Local learning-based adaptive soft sensor for catalyst activation prediction, AIChE Journal 57 (5) (2011) 1288–1301.

[12] N. Lu, Y. Yang, F. Gao, F. Wang, Multirate dynamic inferential modeling for multivariable processes, Chemical Engineering Science 59 (4) (2004) 855–864.

[13] L. Xie, H. Yang, B. Huang, Fir model identification of multirate processes with random delays using em algorithm, AIChE Journal 59 (11) (2013) 4124–4132.

[14] R. R. Andridge, R. J. A. Little, A review of hot deck imputation for survey non-response, International Statistical Review 78 (1) (2010) 40–64.

[15] S. DeSarbo, P. E. Green, J. D. Carroll, An alternating least-squares procedure for estimating missing preference data in product concept testing, Decision Sciences 17 (2) (1986) 163–185.

[16] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, M. Martín, L. Franco, Missing data imputation using statistical and machine learning methods in a real breast cancer problem, Artificial Intelligence in Medicine 50 (2) (2010) 105–115.

[17] C. K. Enders, A primer on maximum likelihood algorithms available for use with missing data, Structural Equation Modeling: A Multidisciplinary Journal 8 (1) (2001) 128–141.

[18] M. B. Richman, T. B. Trafalis, I. Adrianto, Missing data imputation through machine learning algorithms, in: S. H. A. Pasini, C. Marzban (Eds.), Artificial Intelligence Methods in the Environmental Sciences, Springer Netherlands, 2009, pp. 153–169.

[19] R. K. Pearson, Outliers in process modeling and identification, IEEE Transactions on Control Systems Technology 10 (1) (2002) 55–63.

[20] L. Davies, U. Gather, The identification of multiple outliers, Journal of the American Statistical Association 8 (423) (1993) 782–792.

[21] H. Liu, S. Shah, W. Jiang, On-line outlier detection and data cleaning, Computers & Chemical Engineering 28 (9) (2004) 1635–1647.

[22] A. D. Bella, L. Fortuna, S. Graziani, G. Napoli, M. G. Xibilia, A comparative analysis of the influence of methods for outliers detection on the performance of data driven models, in: IEEE Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007, 2007, pp. 1–5.

[23] I. Ben-Gal, Outlier detection, in: O. Maimon, L. Rockach (Eds.), Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic Publishers, 2005, pp. 1–16.

[24] I. T. Jolliffe, Principal Component Analysis, Springer, 2002.

[25] K. Warne, G. Prasad, S. Rezvani, L. Maguire, Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion, Engineering Applications of Artificial Intelligence 17 (8) (2004) 871–885.

[26] K. I. Penny, I. T. Jolliffe, A comparison of multivariate outlier detection methods for clinical laboratory safety data, Journal of the Royal Statistical Society: Series D (The Statistician) 50 (3) (2001) 295–307.

[27] L. Fortuna, S. Graziani, A. Rizzo, M. G. Xibilia, Comparison of soft-sensor design methods for industrial plants using small data sets, IEEE Transactions on Instrumentation and Measurement 58 (8) (2009) 2444–2451.

[28] A. K. Pani, H. K. Mohanta, A survey of data treatment techniques for soft sensor design, Chemical Product and Process Modeling 6 (1) (2011) 1–21.

[29] F. Souza, R. Araújo, Variable and time-lag selection using empirical data, in: Proc. 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011), Toulouse, France, 2011, pp. 1–8.

[30] D. Gabriel, T. Matias, J. C. Pereira, R. Araújo, Predicting gas emissions in a cement kiln plant using hard and soft modeling strategies, in: Proc. 2013 IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2013), 2013, pp. 1–8.

[31] O. Nelles, Nonlinear System Identification - From Classical Approaches to Neural Networks and Fuzzy Models, Springer, 2001.

[32] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, London, UK, 1961.

[33] P. Eshghi, Dimensionality choice in principal components analysis via cross-validatory methods, Chemometrics and Intelligent Laboratory Systems 130 (0) (2014) 6–13.

[34] D.-J. Choi, H. Park, A hybrid artificial neural network as a software sensor for optimal control of a wastewater treatment process, Water Research 35 (16) (2001) 3959–3967.

[35] E. Zamprogna, M. Barolo, D. E. Seborg, Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis, Journal of Process Control 15 (1) (2005) 39–52.

[36] B. Lin, B. Recke, J. K. H. Knudsen, S. B. Jorgensen, A systematic approach for soft sensor development, Computers & Chemical Engineering 31 (5-6) (2007) 419–425.

[37] C. M. Bishop, Neural Networks for Pattern Recognition, Springer, 1995.

[38] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324.

[39] I. Guyon, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[40] M. R. Delgado, E. Y. Nagai, L. V. R. Arruda, A neuro-coevolutionary genetic fuzzy system to design soft sensors, Soft Computing 13 (5) (2009) 481–495.

[41] J. C. B. Gonzaga, L. A. C. Meleiro, C. Kiang, R. M. Filho, Ann-based soft-sensor for real-time process monitoring and control of an industrial polymerization process, Computers & Chemical Engineering 33 (1) (2009) 43–49.

[42] T. M. Cover, J. A. Thomas, Elements of Information Theory, Wiley, 1991.

[43] B. Frénay, G. Doquire, M. Verleysen, Is mutual information adequate for feature selection in regression?, Neural Networks 48 (0) (2013) 1–7.

[44] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Phys. Rev. E 69 (6) (2004) 066138.

[45] J. Beirlant, E. J. Dudewicz, L. Gyorfi, E. C. z van der Meulen, Nonparametric entropy estimation: An overview, International Journal of Mathematical and Statistical Sciences 6 (1997) 17–39.

[46] J. Walters-Williams, Y. Li, Estimation of mutual information: A survey, in: P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, G. Wang (Eds.), Rough Sets and Knowledge Technology, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 389–396.

[47] F. Rossi, A. Lendasse, D. François, V. Wertz, M. Verleysen, Mutual information for the selection of relevant variables in spectrometric nonlinear modelling, Chemometrics and Intelligent Laboratory Systems 80 (2) (2006) 215–226.

[48] D. François, F. Rossi, V. Wertz, M. Verleysen, Resampling methods for parameter-free and robust feature selection with mutual information, Neurocomputing 70 (2007) 1276–1288.

[49] O. Ludwig, U. Nunes, R. Araújo, L. Schnitman, H. A. Lepikson, Applications of information theory, genetic algorithms, and neural models to predict oil flow, Communications in Nonlinear Science and Numerical Simulation 14 (7) (2009) 2870–2885.

[50] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (4) (1994) 537–550.

[51] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.

[52] K. S. Balagani, V. V. Phoha, On the feature selection criterion based on an approximation of multidimensional mutual information, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1342–1343.

[53] P. A. Estévez, M. Tesmer, C. A. Perez, J. M. Zurada, Normalized mutual information feature selection, IEEE Transactions on Neural Networks 20 (2) (2009) 189–201.

[54] F. Souza, P. Santos, R. Araújo, Variable and delay selection using neural networks and mutual information for data-driven soft sensors, in: Proc. 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010), 2010, pp. 1–8.

[55] H.-J. Xing, B.-G. Hu, Two-phase construction of multilayer perceptrons using information theory, IEEE Transactions on Neural Networks 20 (4) (2009) 715–721.

[56] R. Grbić, D. Sližković, P. Kadlec, Adaptive soft sensor for online prediction and process monitoring based on a mixture of gaussian process models, Computers & Chemical Engineering 58 (0) (2013) 84–97.

[57] K. Fujiwara, H. Sawada, M. Kano, Input variable selection for pls modeling using nearest correlation spectral clustering, Chemometrics and Intelligent Laboratory Systems 118 (0) (2012) 109–119.

[58] Y.-H. Chu, Y.-H. Lee, C. Han, Improved quality estimation and knowledge extraction in a batch process by bootstrapping-based generalized variable selection, Industrial & Engineering Chemistry Research 43 (11) (2004) 2680–2690.

[59] X. Wang, M. Han, J. Wang, Applying input variables selection technique on input weighted support vector machine modeling for bof endpoint prediction, Engineering Applications of Artificial Intelligence 23 (6) (2010) 1012–1018.

[60] H. Kaneko, K. Funatsu, A new process variable and dynamics selection method based on a genetic algorithm-based wavelength selection method, AIChE Journal 58 (6) (2012) 1829–1840.

[61] S. Chatterjee, A. Bhattacherjee, Genetic algorithms for feature selection of image analysis-based quality monitoring model: An application to an iron mine, Engineering Applications of

Artificial Intelligence 24 (5) (2011) 786–795.

[62] M. Arakawa, Y. Yamashita, K. Funatsu, Genetic algorithm-based wavelength selection method for spectral calibration, Journal of Chemometrics 25 (1) (2011) 10–19.

[63] H. Kaneko, K. Funatsu, Nonlinear regression method with variable region selection and application to soft sensors, Chemometrics and Intelligent Laboratory Systems 121 (0) (2013) 26–32.

[64] G. Liu, D. Zhou, H. Xu, C. Mei, Model optimization of svm for a fermentation soft sensor, Expert Systems with Applications 37 (4) (2010) 2708–2713.

[65] J. J. Macias-Hernandez, P. Angelov, X. Zhou, Soft sensor for predicting crude oil distillation side streams using evolving takagi-sugeno fuzzy models, in: Proc. IEEE International Conference on Systems, Man and Cybernetics, 2007, pp. 3305–3310.

[66] E. Romero, J. M. Sopena, Performing feature selection with multilayer perceptrons, IEEE Transactions on Neural Networks 19 (3) (2008) 431–441.

[67] R. May, G. Dandy, H. Maier, Review of input variable selection methods for artificial neural networks, in: P. K. Suzuki (Ed.), Artificial Neural Networks - Methodological Advances and Biomedical Applications, InTech, 2011, pp. 19–44.

[68] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.

[69] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67 (2) (2005) 301–320.

[70] I. E. Frank, J. H. Friedman, A statistical view of some chemometrics regression tools, Technometrics 35 (2) (1993) 109–135.

[71] T. Similä, J. Tikka, Combined input variable selection and model complexity control for nonlinear regression, Pattern Recognition Letters 30 (3) (2009) 231–236.

[72] N. Chapados, Y. Bengio, Input decay: Simple and effective soft variable selection, in: Proc. International Joint Conference on Neural Networks (IJCNN'01), Vol. 2, 2001, pp. 1233–1237.

[73] M. Gevrey, I. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models, Ecological Modelling 160 (3) (2003) 249–264.

[74] I.-C. Yeh, W.-L. Cheng, First and second order sensitivity analysis of mlp, Neurocomputing 73 (10-12) (2010) 2225–2233, subspace Learning / Selected papers from the European Symposium on Time Series Prediction.

[75] G. D. Garson, Interpreting neural-network connection weights, AI Expert 6 (4) (1991) 46–51.

[76] Y. Dimopoulos, P. Bourret, S. Lek, Use of some sensitivity criteria for choosing networks with good generalization ability, Neural Processing Letters 2 (6) (1995) 1–4.

[77] I. Dimopoulos, J. Chronopoulos, A. Chronopoulou-Sereli, S. Lek, Neural network models to study relationships between lead concentration in grasses and permanent urban descriptors in athens city (greece), Ecological Modelling 120 (2-3) (1999) 157–165.

[78] V. Lemaire, R. Féraud, Driven forward features selection: A comparative study on neural networks, in: I. King, J. Wang, L.-W. Chan, D. Wang (Eds.), Neural Information Processing, Vol. 4233 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 693–702.

[79] G. Castellano, A. M. Fanelli, Variable selection using neural-network models, Neurocomputing 31 (1-4) (2000) 1–13.

[80] J.-B. Yang, C.-J. Ong, Feature selection using probabilistic prediction of support vector regression, IEEE Transactions on Neural Networks 22 (6) (2011) 954 –962.

[81] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (2002) 389–422.

[82] A. Rakotomamonjy, Analysis of svm regression bounds for variable ranking, Neurocomputing 70 (2007) 1489–1501.

[83] C.-J. Lin, R. C. Weng, Simple probabilistic predictions for support vector regression, Tech. rep., National Taiwan University (2004).

[84] S. Bhartiya, J. R. Whiteley, Development of inferential measurements using neural networks, ISA Transactions 40 (4) (2001) 307–323.

[85] C. L. Mallows, Comments on cp, Technometrics 15 (4) (1973) 661–675.

[86] S. J. Qin, Neural networks for intelligent sensors and control - practical issues and some solutions (1996).

[87] F. Souza, R. Araújo, T. Matias, J. Mendes, A multilayer-perceptron based method for variable selection in soft sensor design, Journal of Process Control 23 (10) (2013) 1371–1378.

[88] T. Marill, D. Green, On the effectiveness of receptors in recognition systems, IEEE Transactions on Information Theory 9 (1) (1963) 11–17.

[89] G. Zahedi, A. Elkamel, A. Lohi, A. Jahanmiri, M. R. Rahimpor, Hybrid artificial neural network-first principle model formulation for the unsteady state simulation and analysis of a packed bed reactor for CO2 hydrogenation to methanol, Chemical Engineering Journal 115 (1-2) (2005) 113–120.

[90] J.-S. R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, 1st Edition, Prentice Hall, 1997.

[91] M.-D. Ma, J.-W. Ko, S.-J. Wang, M.-F. Wu, S.-S. Jang, S.-S. Shieh, D. S.-H. Wong, Development of adaptive soft sensor based on statistical identification of key variables, Control Engineering Practice 17 (9) (2009) 1026–1034.

[92] M. A. Shoorehdeli, M. Teshnehlab, A. K. Sedigh, Training anfis as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended kalman filter, Fuzzy Sets and Systems 160 (2009) 922–948.

[93] J. Mendes, F. Souza, R. Araújo, N. Gonçalves, Genetic fuzzy system for data-driven soft sensors, Applied Soft Computing 12 (10) (2012) 3237–3245.

[94] J. Mendes, S. Pinto, R. Araújo, F. Souza, Evolutionary fuzzy models for nonlinear identification, in: 17th IEEE Conference on Emerging Technologies Factory Automation (ETFA 2012), 2012, pp. 1–8.

[95] S. Soares, R. Araújo, P. Sousa, F. Souza, Design and application of soft sensor using ensemble methods, in: Proc. 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011), Toulouse, France, 2011, pp. 1–8.

[96] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artificial Intelligence 137 (1-2) (2002) 239–263.

[97] S. Soares, C. H. Antunes, R. Araújo, Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development, Neurocomputing 121 (0) (2013) 498–511.

[98] Y. Liu, X. Yao, T. Higuchi, Evolutionary ensembles with negative correlation learning, IEEE Transactions on Evolutionary Computation 4 (4) (2000) 380–387.

[99] M. Matzopoulos, Dynamic process modeling: Combining models and experimental data to solve industrial problems, in: M. C. Georgiadis, J. R. Banga, E. N. Pistikopoulos (Eds.), Process Systems Engineering, Wiley-VCH Verlag GmbH & Co. KGaA, 2010, pp. 1–33.

[100] F. Wang, S. Tan, J. Peng, Y. Chang, Process monitoring based on mode identification for multi-mode process with transitions, Chemometrics and Intelligent Laboratory Systems 110 (1) (2012) 144–155.

[101] J. Yu, Online quality prediction of nonlinear and non-gaussian chemical processes with shifting dynamics using finite mixture model based gaussian process regression approach, Chemical Engineering Science 82 (0) (2012) 22–30.

[102] P. Facco, F. Doplicher, F. Bezzo, M. Barolo, Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process, Journal of Process Control 19 (3) (2009) 520–529.

[103] J. Camacho, J. Picó, Online monitoring of batch processes us-

ing multi-phase principal component analysis, Journal of Process Control 16 (10) (2006) 1021–1035.

[104] N. Lu, F. Gao, Stage-based process analysis and quality prediction for batch processes, Industrial & Engineering Chemistry Research 44 (10) (2005) 3547–3555.

[105] F. A. A. Souza, R. Araújo, Mixture of partial least squares experts and application in prediction settings with multiple operating modes, Chemometrics and Intelligent Laboratory Systems 130 (2014) 192–202.

[106] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1) (1991) 79–87.

[107] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (6) (1974) 716–723.

[108] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6 (2) (1978) 461–464.

[109] T. S. Soderstrom, P. G. Stoica, System Identification, Prentice Hall, 1989.

[110] S. A. Billings, H. B. Jamaluddin, S. Chen, Properties of neural networks with applications to modelling non-linear dynamical systems, International Journal of Control 55 (1) (1992) 193–224.

[111] P. Kadlec, R. Grbic, B. Gabrys, Review of adaptation mechanisms for data-driven soft sensors, Computers & Chemical Engineering 35 (1) (2011) 1–24.

[112] I. Zliobaite, Learning under concept drift: an overview, CoRR abs/1010.4784.

[113] A. Tsymbal, The problem of concept drift: Definitions and related work, Tech. rep., Department of Computer Science, Trinity College: Dublin, Ireland (2004).

[114] L. I. Kuncheva, I. Žliobaitė, On the window size for classification in changing environments, Intelligent Data Analysis 13 (2009) 861–872.

[115] M. W. Lee, J. Y. Joung, D. S. Lee, J. M. Park, S. H. Woo, Application of a moving-window-adaptive neural network to the modeling of a full-scale anaerobic filter process, Industrial & Engineering Chemistry Research 44 (11) (2005) 3973–3982.

[116] M. Liukkonen, E. Hälikkä, T. Hiltunen, Y. Hiltunen, Adaptive soft sensor for fluidized bed quality: Applications to combustion of biomass, Fuel Processing Technology 105 (2013) 46–51.

[117] P. Angelov, A. Kordon, Adaptive inferential sensors based on evolving fuzzy models, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40 (2) (2010) 529–539.

[118] G. Cauwenberghs, T. Poggio, Incremental and decremental support vector machine learning, in: Advances in Neural Information Processing Systems (NIPS'00), 2000, pp. 409–415.

[119] H. Kaneko, K. Funatsu, Adaptive soft sensor based on online support vector regression and bayesian ensemble learning for various states in chemical plants, Chemometrics and Intelligent Laboratory Systems 137 (2014) 57 – 66.

[120] H. Wang, P. Li, F. Gao, Z. Song, S. X. Ding, Kernel classifier with adaptive structure and fixed memory for process diagnosis, AIChE Journal 52 (10) (2006) 3515–3531.

[121] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, J. Vandewalle, Least Squares Support Vector Machiness, World Scientific, 2002.

[122] J.-B. Yang, K.-Q. Shen, C.-J. Ong, X.-P. Li, Feature selection for mlp neural network: The use of random permutation of probabilistic outputs, IEEE Transactions on Neural Networks 20 (12) (2009) 1911–1922.

[123] B. S. Dayal, J. F. MacGregor, Recursive exponentially weighted pls and its applications to adaptive control and prediction, Journal of Process Control 7 (3) (1997) 169–179.

[124] K. Helland, H. E. Berntsen, O. S. Borgen, H. Martens, Recursive algorithm for partial least squares regression, Chemometrics and Intelligent Laboratory Systems 14 (1-3) (1992) 129–137.

[125] T. Komulainen, M. Sourander, S. L. J. Jounela, An online application of dynamic pls to a dearomatization process, Computers & Chemical Engineering 28 (12) (2004) 2611–2619.

[126] C. Li, H. Ye, G. Wang, J. Zhang, A recursive nonlinear pls algorithm for adaptive nonlinear process modeling, Chemical Engineering & Technology 28 (2005) 141–152.

[127] S. Mu, Y. Zeng, R. Liu, P. Wu, H. Su, J. Chu, Online dual updating with recursive pls model and its application in predicting crystal size of purified terephthalic acid (pta) process, Journal of Process Control 16 (6) (2006) 557–566.

[128] O. Haavisto, H. Hyötyniemi, Recursive multimodel partial least squares estimation of mineral flotation slurry contents using optical reflectance spectra, Analytica Chimica Acta 642 (1-2) (2009) 102–109, papers presented at the 11th International Conference on Chemometrics in Analytical Chemistry - CAC 2008.

[129] P. Facco, F. Bezzo, M. Barolo, Nearest-neighbor method for the automatic maintenance of multivariate statistical soft sensors in batch processing, Industrial & Engineering Chemistry Research 49 (5) (2010) 2336–2347.

[130] R. Muradore, P. Fiorini, A pls-based statistical approach for fault detection and isolation of robotic manipulators, IEEE Transactions on Industrial Electronics 59 (8) (2012) 3167–3175.

[131] L. Nan-Ying, H. Guang-Bin, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Transactions on Neural Networks 17 (6) (2006) 1411–1423.

[132] F. Souza, R. Araújo, Online mixture of univariate linear regression models for adaptive soft sensors, IEEE Transactions on Industrial Informatics PP (99) (2014) 1–9.

[133] Y. Fu, H. Su, Y. Zhang, J. Chu, Adaptive soft-sensor modeling algorithm based on fcmisvm and its application in px adsorption separation process, Chinese Journal of Chemical Engineering 16 (5) (2008) 746–751.

[134] P. M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Transactions on Computers C-26 (9) (1977) 917–922.

[135] A. W. Whitney, A direct method of nonparametric measurement selection, IEEE Transactions on Computers C-20 (9) (1971) 1100–1103.

[136] S. D. Stearns, On selecting features for pattern classifiers, in: Proceedings of the 3rd International Conference on Pattern Recognition (ICPR 1976), Coronado, CA, 1976, pp. 71–75.

[137] P. Pudil, J. Novovicová, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (11) (1994) 1119–1125.

[138] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33 (1) (2000) 25–41.

[139] J. H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, USA, 1992.

[140] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 26 (1) (1996) 29–41.

[141] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.