Journal of Computer Engineering
ISSN 10003428

Original Research Article
Vol 13, Issue 7, 2024 | ISSN: 1000-3428

Designing Soft Sensors for the Future Internet: A Comprehensive Survey of Input Selection Methods

Alessandro Bianchi, Lorenzo Ricci, and Elisa Zanon

Department of Electrical Engineering, Politecnico di Milano, 20133 Milano, Italy

Abstract: Soft Sensors (SSs) are inferential models used in many industrial fields. They allow for real-time estimation of hard-to-measure variables as a function of available data obtained from online sensors. SSs are generally built using industries historical databases through data-driven approaches. A critical issue in SS design concerns the selection of input variables, among those available in a candidate dataset. In the case of industrial processes, candidate inputs can reach great numbers, making the design computationally demanding and leading to poorly performing models. An input selection procedure is then necessary. Most used input selection approaches for SS design are addressed in this work and classified with their benefits and drawbacks to guide the designer through this step.

1. Introduction

When dealing with industrial processes, many variables are monitored through online sensors. Some of these variables can be very hard to measure though or can be measured only sporadically due to high cost sensors or lack of the latter. In some cases, variables are measured with high delays because of slow hardware sensors or laboratory analysis, leading to the impossibility of real-time monitoring of the process. Inferential models can then be created to estimate these hard-to-measure variables on the basis of online measured ones. Such models are referred to as Soft Sensors (SSs), or Virtual Sensors [1,2]. Their working principle is summarized in Figure 1.

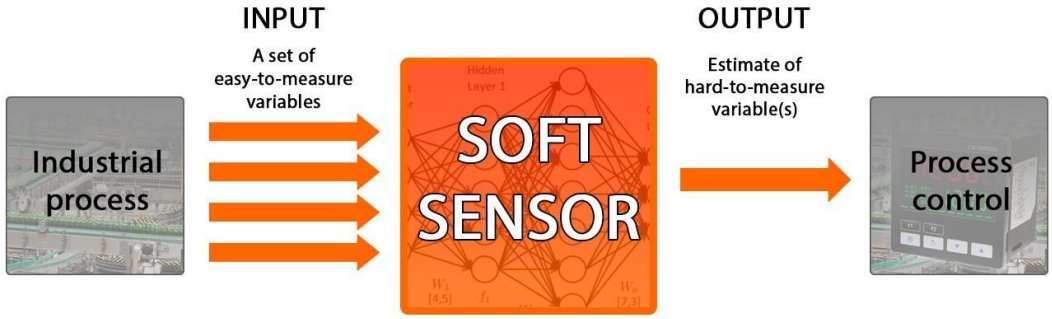

Figure 1. Basic working principle of a Soft Sensor (SS).

SSs were originally exploited in the science of chemometrics, a discipline that studies how to extract information from data sets of chemical systems by using multivariate statistics, mathematics *Future Internet* **2020**, *12*, 97; doi:10.3390/fi12060097

How to cite: Alessandro Bianchi, Lorenzo Ricci, and Elisa Zanon (2024). Designing Soft Sensors for the Future Internet: A Comprehensive Survey of Input Selection Methods. Journal of Computer Engineering, 13(7), 109–140. DOI:
https://doi.org/10.5281/zenodo.19352812

www.mdpi.com/journal/futureinternet and computer science [3,4]. Chemometrics solves prediction problems by learning models from data and exploiting machine learning, system identification, artificial intelligence and statistical learning theory [1,2,5].

SSs' industrial use ranges over a number of different types of processes, such as refineries [6–11], chemical plants [12], cement kilns [13,14], power plants [15,16], pulp and paper mills [17,18], food processing [2,19], nuclear plants [20,21], pollution monitoring [22], polymerization processes [23–25] or wastewater treatment systems [26–29], just to mention a few. In Figure 2, a distillation column process from a refinery is shown through its control software: displayed online measurements are collected in a historical database for future use.

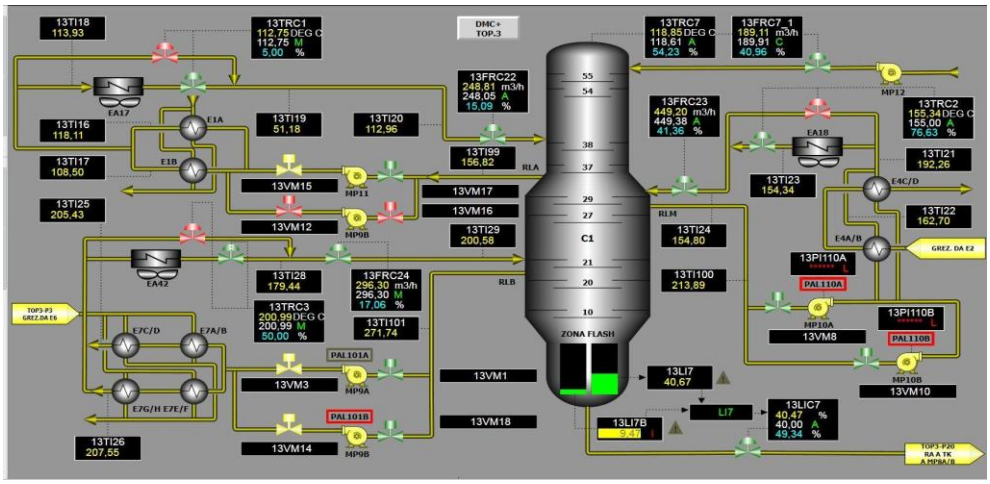

Figure 2. Control software of a real distillation column from a refinery, displaying a part of the available easy-to-measure variables measured by online sensors.

SSs' real-time estimation capability allows developing tight control policies and since they represent a low-cost alternative to expensive hardware devices, SSs allow the realization of more comprehensive monitoring networks. The real-time estimation obtained by the SS can be used by a controller, while the corresponding delayed measurements allow the model to be improved, by avoiding an error propagation effect. Besides their use for plant control, they are used to approach a number of other different problems as well, such as measuring system back-up, what-if analysis, sensor validation and fault diagnosis [7,30–41].

The back-up of measuring instrumentation consists of substituting unavailable measuring equipment, to avoid degradation of plant performance and rises in costs. This can become necessary since measuring devices, and their data transmission systems, generally face harsh working environments that impose periodic maintenance procedures or lead to faults. Therefore, when a maintenance intervention is performed, measuring hardware needs to be turned off and suitably substituted throughout the entire process.

What-if analysis consists of using the model to perform simulations of the system dynamics with respect to types of input that are of interest, for a given time span, with the aim of both achieving a deeper understanding of the system behavior or designing suitable control policies.

SSs allow to reduce the need for measuring devices as well, improving system reliability and decreasing sensors acquisition and maintenance costs. They can eventually work in parallel with hardware sensors, giving this way useful information for fault detection tasks and they can be easily implemented on existing hardware and retuned when system parameters change. The main goals of fault detection are to perform early detection of faults providing as much information as possible about it, to provide a support system for scheduled maintenance interventions, and to provide a basis for the development of fault-tolerant systems.

SSs can be built with different approaches. Since they are dynamic I/O models, simpler ones, like linear models, are usually preferred because of the lower time and computing demand, and if a priori physical knowledge of the process to model is given, a white-box design approach is possible as well. However, in an industrial environment, because of the complexity of the processes and the amount of available data, nonlinear models are needed and data-driven black-box approaches lead to satisfactory results. Available data must be representative of the dynamics of the process, and the choice of the right inputs is, for this reason, a crucial step in the design process.

Input selection is a widely addressed subject. Several surveys on the topic are reported in literature. In [42], a taxonomy of the most used methods for Artificial Neural Network models is given, as well as in [43] where a review of the approaches used on microarray data for Support Vector Machine (SVM) models is performed. In [5], a review of the SS design in its entirety is given. In [44], filter selection methods exploiting Mutual Information are reviewed, while [45] provides a survey of wrapper classified feature selection methods. The work in [46] focuses on feature selection methods of the semi-supervised class. This paper actually focuses on the input selection step of SS design independently from the specific model adopted by the designer, with the aim to help in the choice of the most suitable technique by exploring all the classes of the state-of-the-art methods.

With the introduction of Industry 4.0 technologies, one underrepresented topic is the one related to the role of humans in manufacturing and how technology can enhance the integration between human and machine. Many manufacturing systems are people-oriented, meaning human operators interact
with intelligent devices around them. In such environments, people are the ones with the responsibility for actions and decisions. In this case, automation aims to supply devices able to collect and aggregate data, so to provide them in a user-friendly way to the person in charge of making the right decision based on the available data. All the design processes require human mediation, and this generally applies to industrial automation and machine learning applications as well. This consciousness gave birth to the human-in-the-loop approach, which puts human knowledge and experience as a pivot of machine learning processes [47,48]. In the SS field, human knowledge of the industrial processes provided by technicians represents a vital resource for the design process. This was shown in [49] where technician experience was crucial in the optimization of the number of inputs to build the best performing SS of a unit distillation process.
As the design steps strictly correlated to each other, each step is firstly explained in Section 2.
The input variable selection problem is addressed in Section 3. The two main classes of approaches, Feature Extraction (FE) and Feature Selection (FS) are respectively discussed in Sections 4 and 5. In the final Section 6, conclusions are drawn as well as a table that summarizes the techniques cited in this work. It sorts the methods by classes along with their advantages and disadvantages, with the intent to provide some guidance to the designer for the one that most accommodates a specific case. In the same section, references were arranged in two tables: one for the methods and one for real-case applications.

2. SS Design Stages

SS design stages follow the typical steps of pattern recognition [50] as well as system identification theory [51]. In an industrial environment, ad-hoc experiments to collect suitable identification datasets are, in general, not possible as in the system identification practice. So, data have to be retrieved from industries historical databases.

SS design steps can be summarized as follows [1]:

1. Data collection and filtering;
2. Input variables selection;
3. Model structure choice; 4. Model identification;
5. Model validation.

Each of these steps is crucial for the good success of the further one.

Data is stored by industries in historical databases, generally provided from a supervisory control and data acquisition (SCADA) control system [52] or a distributed control system (DCS) [53]. Collected data must be capable of representing the whole dynamics of the system, since the model cannot provide more information than the one stored in the data itself. After data are collected, the first stage of the design regards its filtering and preprocessing [1,2]. This is due to unprocessed data from databases coming with well-known problems, such as oversampling, outliers and missing data and accuracy problems, such as offsets, seasonal effects and high-frequency noise. Therefore, the designer should carefully deal with them and prepare data to become suitable for the next designing steps [54–56]. Common preprocessing stages consist of resampling, noise filtering, outlier detection and removal and normalization.

Data collected in plant databases usually come with different sample rates. Easy-to-measure variables are automatically measured with online available sensors; while hard-to-measure ones cannot be measured automatically, but more commonly only sporadically, at high cost and with high delays, such as in the case of laboratory analysis [1,57]. For this reason, the former usually present high sample rates, even higher than the one required by the sampling theorem, while the latter tend to be downsampled [58,59]. High sample rate can lead to huge datasets that can suffer from data collinearity. Therefore, resampling becomes necessary to synchronize the variables back together [60] and to avoid dealing with huge datasets.

Missing data and outliers are quite common problems in databases collected from industries. The former occur when values are missing in the observation of a variable; the latter are actually inconsistent data with the majority of the recorded ones that greatly deviate from the typical range of values, such as peaks, saturations, flat trends and discontinuities. They can both be caused by sensor or process faults and measurement noise. They are usually handled by removing the samples that contain them or by filling the missing observations with some imputing method. Outlier detection is, however, a tough task achieved through statistical techniques, such as the 3σ-rule, and the final validation has to be manually performed by a plant expert to avoid outlier masking (a false negative) and outlier swamping (false positive) [61].

High frequency noise is generally induced by measurement instruments and can be filtered out with low-pass filters. The appropriate bandwidth is chosen by using spectral analysis.

Once data have been preprocessed, selection of input variables is one critical step in the SS designing process. The importance of this stage is addressed in the next section along with the main techniques adopted.

The further step consists of the choice of the model structure, which is based on the a priori knowledge of the system. Mechanistic (or white-box) models are the ones obtained on the basis of first principles analysis, such approach requires a deep physical knowledge of the process. However, due to the complex processes occurring in industrial plants and given the great amount of collected data, the use of gray- or black-box data-driven identification approaches can be a good choice, since satisfactory results can be achieved with reasonable computational and time efforts. Such data-driven models are only based on the empirical observations of the process and generally require slight knowledge of the system. A great

work of data processing is needed anyway. Since it is hard to find a general solution equally satisfactory for any case, any plant experts' knowledge must still be taken into account. It can regard the input variables importance, the system order, the operating range, time delay, degree of nonlinearity or sampling times. Linear models are usually preferable as they are computationally easier, since the numerical procedures and the design of a controller are simpler. Such approximation is possible when certain conditions are met, like small variations around the nominal working point, a small degree of nonlinearity of the process or what degree of approximation is needed for the model. If a linear model does not show good results, a nonlinear identification approach is needed. In the industrial field, parametric structures such as FIR, ARX or ARMAX are widely used in both the linear and nonlinear (NFIR, NARX, NARMAX, respectively) cases [1,62,63], as well as static models.
Industrial SSs are generally designed with Artificial Neural Networks (ANN), mainly with the
Multi-Layer Perceptron (MLP) structure [7,11,63,64], Convolutional Neural Networks (CNN) [65,66],
Generative Adversarial Networks (GAN) [67,68], Radial Basis Networks [69], Wavelet Networks [70], Hinging Hyperplanes [71], Deep Belief Networks [72,73], Stacked Autoencoders [74], Long Short-Term
Memory Networks [75], Support Vector Regression [76], Gaussian Processes Regression [77], Extreme Learning Machines [78], Fuzzy Systems and Neuro-Fuzzy Systems [79,80], just to mention a few [5,81–88]. In some cases, the designer can choose to create more linear or nonlinear models for the same system, each one for a different working point, instead of a single one covering all of the system dynamics. In such cases, the models are aggregated by a suitable algorithm such as fuzzy logic or neural stacking. Such approaches are called ensemble methods [88].

In some cases, the designer could deal with systems showing finite time delay between the input variables and the process output, sometimes caused by the measurement process. Several approaches are proposed in literature. In [57], a FIR model along with an Expectation–Maximization (EM)-based algorithm is used to estimate the model parameters and the time delays. In [79], a Takagi–Sugeno-fuzzy model is tuned using a genetic-algorithm-based approach to identify delays. Genetic Algorithms have also been used to estimate the time delays as in [89,90], where they are used to minimize the Joint Conditional Entropy between the input and output variables.

In [91], the problem of selecting input time-lags is treated as a variable selection problem with a multidimensional mutual information estimator. Mutual Information is also applied to delay selection in the design of a SS in [92,93], where the delay is estimated by using the Normalized Mutual Information and delayed replicas of the inputs. In [10], delays are estimated through the learning phase of a Deep Belief Network.

Once the structure has been chosen, the whole preprocessed and selected data set should be partitioned in subsets for the last two design steps, as:

- Identification data
- Validation data

The first allows to identify the candidate models and empirically estimate their unknown parameters.

Finally, the validation step exploits validation data to verify whether the model is able to adequately represent the system and perform generalization to new samples. In SS design, as in pattern recognition and system identification, it is important to perform the validation on different data with respect to the ones used for the model identification. This is particularly done to investigate overfitting phenomena. Validation techniques analyze the

model residuals characteristics by looking for any undesired correlation between them and present and/or delayed samples of model inputs and outputs. This can be immediately performed through graphical techniques such as visual comparison of the time-plotted output of the system and of the one estimated by the model, lag plots, correlation graphs or histograms. Other performance metrics usually adopted are the Mean Squared Error (MSE), the Normalized Root Mean Square Error (NRMSE), the Mean Absolute Error (MAE), Akaike's Final Prediction Error (FPE), Akaike's Information Criterion (AIC) [94,95], Rissanen's Minimum Description Length (MDL) [96], Bayesian Information Criterion [97], C_P statistics [98]. When dealing with small datasets, cross-validation techniques such as the K-fold cross validation or the leave-one-out (LOOCV) one are employed. They consist of splitting given data samples in K number of groups (or folds). At each iteration, one of the K groups is used for validation, while the other K-1 groups are for training. This is done for all the K groups, and the final performance is given as an average of the performance measured at each iteration. LOOCV consists of the same approach when K is the number of samples as well.

A scheme of the design process of an SS is given in Figure 3.

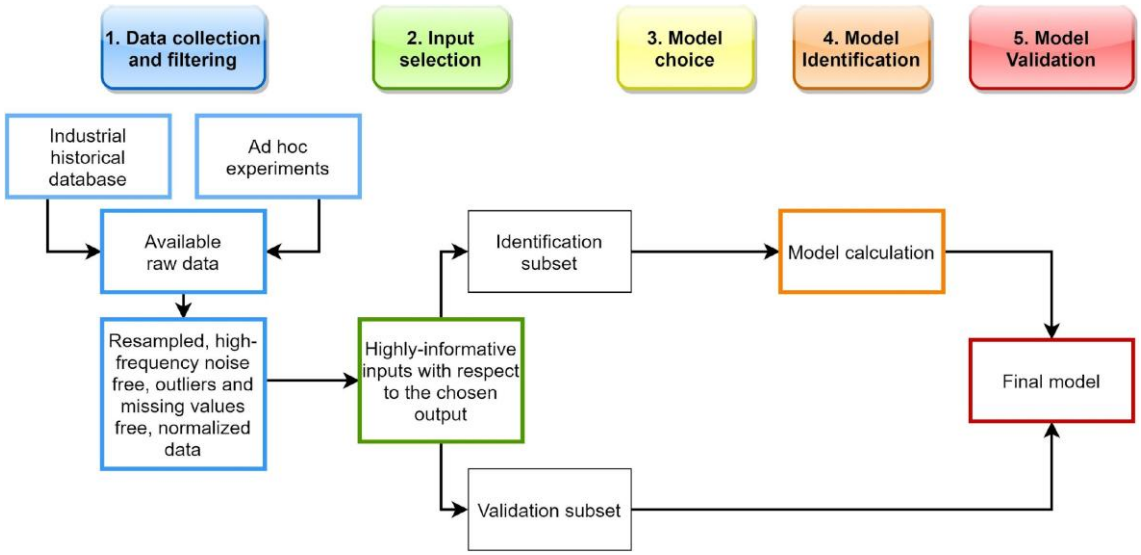

Figure 3. Main steps in SS design.

3. The Input Selection Problem in SS Design

The model design takes for granted that at least one or more of the candidate inputs is able to describe the output of the system chosen by the designer. If that is not the case, the model development is an impossible task and the available data should be reconsidered: ad hoc experiments should be performed to cover the dynamics of the system or a different variable can be chosen as output. Generally, given the initial set of candidate inputs, it is common to have irrelevant ones or to have correlations between some of the input variables, making them redundant. Irrelevant inputs are those that have little or no predictive power with respect to the output, so they can be discarded without losing information. The concept of redundancy is instead associated with the level of dependency among two or more variables.

The optimal subset of input variables is then unknown. What the designer wants to achieve is to discard such inputs, reducing in this way the degree of redundancy and to remove no informative variables, with the aim of detecting the relevant high informative ones to build an optimal set.

The reason why the number of inputs is reduced, is because the dimensionality and the representability of the input space is one of the factors that may limit the successful design

of an SS. In the case of industrial processes, candidate inputs can reach great numbers [91,99]. Moreover, if in the model structure choice a non steady-state type of model (such as the ones mentioned) is preferred, the number of candidate variables is multiplied by the model order, making the number of variables even larger, mostly in the case of strong persistence systems.

When this occurs, a large number of inputs dramatically increases the computational cost of the model identification step [100] and leads to a large number of model parameters to be estimated, generally causing poor generalization and high probability of overfitting [101]. High-dimensional datasets that suffer the so-called "large p, small n" problem (where p is the dimension of the input space and n is the number of samples), tend to be indeed affected by overfitting. A model suffering overfitting mistakes small fluctuations for important variance leading to errors on test data. This unavoidably increases in the presence of noisy measurements. The reason behind this phenomenon is called curse of dimensionality [102]: as the input dimensionality increases, the volume of the space increases so fast that the available data become sparse, meaning that the amount of input samples needed to support the result grows exponentially with its dimensionality [103].

On the other hand, dimensionality reduction shortens the model development time, improves the predictor performance, facilitates data visualization and data understanding. Also, a reduction of the number of variables implies a lower number of required hardware sensors, decreasing costs associated with them, as well as fewer missing data and outliers to deal with.

The objective is therefore to find the input subset of minimum cardinality that preserves the information contained in the whole initial set with respect to the output; or put in other words, the subset containing the fewest inputs required to properly describe the behavior of the output. To deal with the problem, approaches can be classified in [104]:

- Feature Extraction (FE, Unsupervised)
- Feature Selection (FS, Supervised)

These two classes of methods are addressed in the next sections. A full taxonomy of the approaches is depicted in Figure 4.

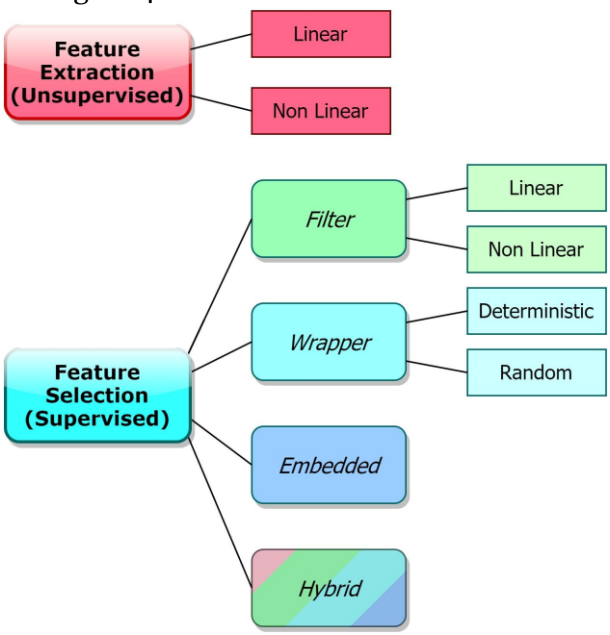

Figure 4. Main classification of input selection methods.

4. Feature Extraction

FE is a class of unsupervised methods that create new features based on transformations or combinations of the original variable set. The most well-known FE algorithm is Principal Component Analysis (PCA) [105]. It uses orthogonal transformation to express a set of p variables as d vectors called principal components, with $d < p$. The model identification is then performed on such found components. PCA finds the first principal component with the largest variance in a latent space where the original input space is projected, using the covariance matrix and its eigenvalues and eigenvectors. All the successive components are the ones with the highest variance that are orthogonal to the others. However, the relationship between the variables is assumed to be linear, such as the one between the principal components and the output. Therefore, the procedure will fail at identifying any nonlinear relationship in the data. Moreover, the transformations of the input variables are done without taking the output variable into account, with the method being unsupervised.

The first problem is overcome by some nonlinear versions of the algorithm, such as Nonlinear PCA (NLPCA) [106] and Kernel PCA (KPCA) [107]. The first one uses Autoassociative Neural Networks to perform the identity mapping: the network inputs are reproduced at the output layer, with an internal bottleneck layer and two additional hidden layers. The second one generalizes linear PCA into the nonlinear case using the kernel method: the original input vectors are mapped into a higher dimensional feature space in which the linear PCA is then calculated. In both cases, the transformations of the data can be highly complex and interpretation of the PCs is a harder task.

The unsupervised limitation led to the introduction of a supervised version of PCA, Supervised-PCA (SPCA) [108], where PCA is applied to a subset of the inputs selected on the basis of their association with the output.

Other PCA variations are Independent Component Analysis (ICA) [109] and Probabilistic PCA (PPCA) [110]. ICA is originally developed to blindly separate multivariate signals with the goal of recovering mutually independent but unknown source signals from their linear mixtures without knowing the mixing coefficients. This is used to linearly transform original inputs into features that are mutually statistically independent. In PPCA, a Gaussian latent factor model is considered and then the PCAs are obtained as the solution of a maximum marginal likelihood problem, where the latent factors are marginalized out.

PCA, SPCA, ICA, KPCA and PPCA have been applied as a way of reducing the dimensionality of the data in many works [28,36,56,89,111–115]. A comparison between PCA, KPCA and ICA as dimensionality reduction methods is performed in [116], where KPCA showed the best performances among the three. In [117], an original feature selection method that combines ICA and false nearest neighbors (FNN) is proposed as ICAFNN.

Another nonlinear dimensionality reduction FE approach is multidimensional scaling (MDS) [118,119] along with its variations such as Principal Coordinates Analysis (PCoA) [120], metric-MDS, non-metric MDS and generalized MDS. MDS is a set of related ordination techniques to display the information of a dataset in a distance matrix that contains the distances between each pair of points of such dataset. The algorithm places each point into a space of a chosen dimension N such that the distances are preserved as well as possible. Other nonlinear FE methods are Isomap and its variations [121–123], Locally Linear Embedding (LLE) [124,125] and Laplacian Eigenmaps [126].

Isomap is a combination of the Floyd–Warshall (F-W) algorithm with classic MDS, where the pair-wise distances are assumed to be only known between neighboring points and the others are computed with the F-W algorithm. Then, classic MDS is performed to compute the reduced-dimensional positions of all the points. LLE has faster optimization and better results than Isomap. LLE also finds a set of the nearest neighbors of each point,

so to describe it as a linear combination of them after computing a set of weights for each neighbor. It then finds the lower-dimensional embedding of the points such that each point is still described with the computed linear combination. Laplacian Eigenmaps adopts spectral techniques to perform dimensionality reduction, based on the assumption that the data lies in a low-dimensional manifold that exists in a higher-dimensional space.

Another unsupervised procedure to produce a low-dimensional representation of an input space is given by Self-Organizing Maps (SOM, or Kohonen maps) [127], a type of artificial neural network used to perform dimensionality reduction. The task is achieved through Vector Quantization (VQ),
which is a classical quantization technique from signal processing that describes a larger set of n vectors by c codebook or prototype vectors. The candidate set of inputs is considered as the prototype vectors of the SOM. Similar candidate variables will be identified by the formation of groups, which have the closest proximity to the same prototype vector. The distance measures generally used to evaluate this proximity are linear correlation or covariance in the linear case, otherwise Mutual Information or Entropy in the nonlinear case.

The drawback of FE techniques is that the variables in the new space do not have any physical meaning and they become difficult to be interpreted. In addition, in the case of industrial processes, since the original inputs are still needed to obtain the projections, the number of required hardware sensors for the estimation is not reduced, hence losing one of the important advantages brought by the use of a fewer number of inputs.

5. Feature Selection

FS refers to a class of supervised methods that select the best subset from the original feature set, retaining this way their physical meaning [128]. The selection of the input variables takes the relationships between inputs and outputs variables into account, either related to the accuracy of the corresponding model or not. Different strategies are used to search among the possible sets of candidate variables and they can be classified in the following groups of methods [45,129]:

- Filters
- Wrappers
- Embedded (model-based)
- Hybrid approaches

Any of these procedures for input selection defines a criterion or a cost function to quantify the quality of a subset as well as a search strategy to determine the candidate subset [50]. This is done since exhaustive search is not recommended or even not feasible in most cases, due to the extremely high computational expensiveness given the number of inputs. In an exhaustive search all the possible combinations of inputs are considered, and therefore, given n candidate input variables, 2^n possible combinations of subsets exist.

So search strategies provide an efficient method to search through the many possible combinations of inputs and can be classified as local, that start their search from a point and then move incrementally, or global, that consider many combinations.

Forward selection and backward elimination are two linear incremental local strategies [130]. Forward selection methods start with an empty input subset and then inputs from the candidate set are included one at a time. The chosen input should be the one that most contributes to the output, according to the criterion the specific method uses. The approach is computationally efficient and results in relatively small input sets, but because of its nature it may encounter a local optimum, terminating prematurely, or may ignore informative combinations of variables that are not very relevant individually [129]. An extension of this

strategy is the step-wise selection in which past input variables may be removed at each iteration to better handle redundancies in the subset.

Backward elimination, as opposed to the previous, starts by first considering all the candidate inputs. Then subsets with one less input are built and examined to evaluate whether the deleted one is more or less significant. The procedure goes on until no more inputs can be deleted, according to the adopted criterion. Such approach is generally more computationally demanding.

In their floating variants (Sequential Forward Floating Selection—SFFS and Sequential Backward Floating Selection—SBFS, respectively) [131], there is an additional inclusion or exclusion step to remove variables once they were included (or excluded), so that a larger number of subsets can be sampled.

A heuristic search involves global strategies that implement a search of random solutions in the search space and increase the focus in regions that lead to good solutions. The nature of the approach allows finding global or near-global optimal solutions. They are usually implemented with evolutionary algorithms as Genetic Algorithms (GA) and Ant Colony Optimization Algorithms (ACO) [132–134] or Simulated Annealing (SA). The approach requires the tuning of search parameters that trade-off the amount of the search space that is explored and the rate at which the final solution is reached.

The same given taxonomy of methods is used for semi-supervised feature selection methods as well, as stated in [46]. Semi-supervised approaches for evaluating input relevance are exploited in cases in which both labeled and unlabeled data are available. This often happens since unlabeled data are more easily accessible than labeled ones, where hard-to-measure variables must be measured and recorded as well as easy-to-measure ones in order to provide enough data to build the predictive model. This paper takes into account only methods for which available data provided to the designer are labeled. The cited work gives, however, a comprehensive detailed survey of input selection methods in a semi-supervised environment.

5.1. Filter Methods

These are methods that select subsets of inputs as a preprocessing step, exploiting statistical measures to quantify the quality of a subset (multivariate methods) or providing a ranking of each single variable based on a relevance index, eventually rejecting those with a value that falls below an established threshold (univariate methods). The relevance measure is usually a bivariate statistical analysis that evaluates each candidate-output relationship, so filter methods are usually characterized by incremental search strategies. In filter methods that operate on each input variable individually, dependencies and interactions between them are disregarded, not accommodating the multicollinearity problem [135]. This is the reason why they are often used as a first screening step, before more sophisticated methods are applied in hybrid approaches.

Filter methods do not require to build any prediction model first since they are independent of the chosen model structure: these approaches separate the inputs selection task from the model identification step. This makes such methods simple and fast, because they are the least computationally demanding ones. They allow for good empirical results even in cases in which the number of samples is smaller than the number of inputs.

The most common filter method consists of the analysis of the correlation coefficient (CC). The most common coefficient is Pearson's correlation coefficient ρ, which is a measure of the linear correlation between two variables, in this case the candidate input and the output [136]. The linear correlation between each input and the output is computed and then a ranking list of the inputs is provided, according to the scores. Practical examples of this approach are given by [1,23,137].

In [138], different coefficients such as Distance Correlation (DC) [139], Maximal Correlation (MC) [140] and Maximal Information Coefficient (MIC) [141] are combined with Pearson's coefficient to introduce a more robust factor that can be generally used when the relationship between the variables is not necessarily linear. Being the dependencies between variables neglected, if there is correlation between the candidate inputs, such approach would select too many variables giving problems of redundancy. To accommodate the problem, partial correlation can be used instead. It measures the strength of the relationship between two variables, while controlling for the effect of one or more other variables that is discounted.

In nonlinear settings, ρ is generally replaced by Mutual Information (MI) [142], a measure of dependence based on information theory and Shannon's notion of entropy that quantifies the information about a variable provided by a second variable. The reason why MI is adopted in nonlinear settings is because it is based on probability distributions within the data and makes no assumption on the structure of the dependence between the variables. It also is insensitive to noise and data transformation, making it a robust measure. In a univariate approach, it provides a ranking like in the linear case [129,143]. In multivariate approaches, when the number of candidate inputs is large, it is not possible in practice to evaluate the MI between all the possible subsets and the output, so incremental greedy procedures are frequently used. These approaches can possibly detect subsets of features that are jointly relevant or redundant. In such a context, probability density functions are unknown in real-world problems and MI has to be estimated. The most adopted methods are Nearest Neighbors-based algorithms that show good results [77,144–146] and are shown to outperform other common estimators such as the histogram one, the kernel estimator and the b-spline estimator [147], as well as the CC approach [91]. The basic histogram method is, however, preferred when dealing with small variables because of its simplicity [90]. When the number of variables to work with increases, multivariate MI methods become complex due to the estimation of the probability density function [148].

The multivariate problem is approximated with a univariate approach in [149], where the Mutual Information Feature Selector (MIFS) is introduced: a heuristic criterion is adopted to find the subset that maximizes MI. MIFS's performance can, however, be degraded as a result of large errors in estimating the mutual information. Another common drawback is the selection of redundant variables if an input is closely related to the already selected one. This is the reason why a new greedy selection method was introduced as MIFS-U (MIFS-Under Uniform Information Distribution) [150]. It is shown that the two algorithms are equivalent to the maximization of multivariate MI [151]. However, they could both lead to the selection of irrelevant variables earlier than relevant ones if the cardinality of the inputs subset becomes big. This is partly solved by mRMR (minimum redundancy-maximum relevance) [152]. The criterion of maximum-relevance ensures that the selected inputs are highly informative by evaluating their high degree of correlation with the output. The criterion of minimum-redundancy looks for inputs that are maximally dissimilar from one another, in order to build the most useful set of relevant variables. In [153], a novel mutual information feature selection method based on the normalization of the maximum relevance and minimum common redundancy (N-MRMCR-MI) is proposed, where the normalization method is applied to the Max-Relevance and Min-Common-Redundancy (MRMCR) criterion and returns a correlation measure that takes values between 0 and 1. NMIFS is another algorithm that proposes the average normalized MI as a measure of redundancy among inputs [154]. In [155], a variable selection method based on Dynamic Mutual Information is proposed and called DMIFS. In [156], a selection based on Partial Mutual Information (PMI) is introduced and successfully applied in other works as well [157,158]. When datasets become extremely large, however, the greedy optimization tends to be infeasible. This can be overcome by the use of parallel computing to speed the procedure

up. In [159], the greedy optimization procedure is revisited to propose a semi-parallel optimization paradigm that works as the other state-of-the-art algorithms, but in a fraction of the time. The algorithm is tested even on a dataset of more than a million candidate inputs. Another method proposed after MIFS is Information Theoretic Subset Selection (ITSS) [160], described as a multivariate MI approach where indications on when the growth of the subset has to be stopped are given, as opposed to the MIFS algorithm. The method exploits a parameter based on MI called Asymmetric Dependency Coefficient (ADC) to estimate the knowledge of the output carried by the selected subset. When the ADC reaches the maximum value of 1, a full knowledge of the output is reached. A review of variable selection methods based on MI is given in [44].

Lipschitz's quotients can be used for input selection by computing the Euclidean distances in the input space and in the output at different time instants [161]. Such approach is based on the continuity property of the nonlinear function representing the input–output model and it depends only on the input–output data collected through experiments. In order to evaluate each subset of variables (or to evaluate the importance of the variable or variables excluded), each Lipschitz's quotient computed for that subset is compared with the one computed for the whole candidate set. However, this approach requires the computation of the quotient for all the possible combinations of the input variables, resulting in a high computational demand [162].

In [163], several linear filter variable selection methods are compared to nonlinear ones using two large databases, in particular a synthetic one and a real-world one. Results showed nonlinear methods to be a generally preferable and more robust tool.

5.2. Wrapper Methods

Such class of methods perform the input variable selection by evaluating the performance of the final model via cross-validation, where each model corresponds to a unique combination of inputs [45]. The assessment is done by using the same criteria that are used to evaluate the predictive performance in the model validation design step, for example, the MSE [164–168]. In the case of the use of the MSE as an optimality criterion, the drawback is that the best model could not be the optimal one, since models with a large number of inputs tend to suffer overfitting. So other criteria like Akaike's Information Criterion (AIC) [94,165,169,170], or the C_P (Mallows' Coefficient) statistics [1] are adopted since these measures penalize overfitting by determining the optimal number of input variables as a trade-off between the model size and the accuracy.

With respect to filter methods, these approaches are slower and computationally and time expensive, since a new model is created every time a new subset is picked. Being the evaluation done on the final model, they generally give better results.

As explained in Section 5, the ideal approach would be evaluating all possible subsets, but as it is infeasible, the use of a search strategy is needed. On the basis of the adopted search algorithm shown in the same section, wrapper methods can be classified as deterministic or randomized [171,172]. Deterministic wrappers use Sequential Feature Selection greedy algorithms like Forward Selection, Backward Elimination and their variants. They generally present a lower overfitting risk [173–175]. Randomized wrapper methods are the ones adopting heuristic search and exploit a randomized criterion in the selection of the subset. As already stated, they have more parameters to be tuned [176,177].

5.3. Embedded Methods

In this case, variable selection depends on the structure and on the type of the used model: a specific characteristic of the model or of its learning process is used to define the criterion. These methods, compared to the filter ones, are slower and give bad generalization

performance (overfitting) when not enough data is available; vice-versa when enough data is available, they generally outperform filter methods [5,42].

Recursive feature elimination (RFE) [129,178,179] is a backward-elimination embedded input selection strategy. RFE consists of an iterative process of training a model, where all the candidate inputs are initially used. At each iteration, RFE seeks to improve generalization performance by removing the least important variable in which the deletion will have the least effect on training error. This method works well for problems with small training samples and high input dimensionality, but it tends to remove redundant and weak variables, keeping independent ones. As already stated in this paper, weak input variables that are useless by themselves can provide a good improvement in performances when combined together, so simply removing them can degrade the classification performance.
For this reason, variations of the algorithm have been proposed such as Enhanced-RFE (EnRFE) [180] or RFE-by-sensitivity-testing (RFEST) [181]. Original RFE does not concern the further state at each iteration, as opposed to EnRFE that will retain redundant or weak features that are useful
when combined with other features. It is shown that EnRFE performs better than its original version. In RFEST, RFE is used with sensitivity analysis to rank inputs and to overcome the same limitations.

Sensitivity analysis [182] is an input selection method in which the model is first trained with all the candidate inputs, then one input is analyzed by measuring the variation of the output when it is perturbed [183–186]. If considered irrelevant by the sensitivity analysis, it is then removed.

Evolutionary ANNs (EANNs) [187] are population-based algorithms for neural network models that simulate the natural evolution of biological systems to optimize the NN and to determine the optimal set of input weights. When the optimization procedure sets an input connection weight close or equal to zero, then that input variable is excluded, making the input selection embedded within the EANN approach.

Least Absolute Shrinkage and Selection Operator (LASSO) [188,189] is a regularization method that provides input selection in an embedded way. Regularization is a method of reducing variance in a linear regression model at the cost of introducing some bias. This is done by adding the model error function of a penalty term. Ridge Regression (RR) [190] penalizes the sum of squared coefficients, the so-called L_2 penalty. When the function is forced to be less than a fixed value, the penalty term shrinks the model coefficients leading to a lower variance and a lower error value. This decreases the complexity of the model but does not reduce the number of variables, it rather just shrinks their effect. LASSO actually penalizes the sum of the parameters absolute values, the so-called L_1 penalty. This makes some of the parameters shrink to zero, which is never the case in ridge regression, eliminating some variables entirely and performing variable selection, by giving a subset of predictors that helps mitigate multicollinearity and model complexity. Elastic Net (EN) [191] linearly combines the L_1 and L_2 penalties from LASSO and RR and can be optimized to effectively perform coefficient shrinkage as well as setting some of them to 0 for sparse variable selection. LASSO regularization for inputs selection is extended to the nonlinear case as well with the name of LASSO-MLP [162,192]. In this case, the L_1 penalty term is added to the error function of a single-layer MLP and then variable selection is performed as in linear LASSO.

5.4. Hybrid Methods

Merging different methods often brings better results and less computational demand. Different combinations of input selection methods and classes can be performed to further reduce the number of inputs. Filter methods can be used, such as the pre-filtering method in

[1], where correlation coefficients and scatter plots are used as a preselection and then partial correlation and Mallows' C_p statistics are used for input selection. In [193], a combination of wrapper and embedded methods are proposed and called SBS-MLP. It presents low computational cost and tends to equally-perform or outperform other state-of-the-art methods it was compared with.

In [194], a selection method is proposed in which Nearest Correlation Spectral Clustering Variable

Selection (NCSCVS), a method that clusters inputs into groups based on the correlation between variables by nearest correlation spectral clustering, is used as a filter step and then integrated with group LASSO. This method is called Nearest Correlation Spectral Clustering Group LASSO (NCSC-GL). In [195], the NC-based method is used to search for inputs correlated with the output, and then che correlation similarity between the inputs and the output is used to weight the respective input in the model. The method is called Nearest Correlation-Based Input Variable Weighting (NCVW).

In [29], a self-organizing map (SOM) is used to reduce the dimensionality of the input space and obtain independent inputs. Then, to determine which inputs have a significant relationship with the output, a hybrid approach exploiting GA with a General Regression Neural Network (GRNN) is proposed and called GAGRNN.

In [83], variable selection is performed by first ranking the candidate inputs exploiting correlation coefficient analysis, then the optimal subset is chosen with a wrapper approach by evaluating the prediction performance of different models.

The gradient-based leave-one-out gene selection (GLGS) algorithm [196] combines a variant of the Leave-One-Out Cross-Validation (LOOCV) with the Gradient Descent Optimization algorithm to PCA, to perform input dimensionality reduction.

In [197], an ensemble input set that maintains informative inputs from the original set is formed as a combination of the output feature set of a population of LASSO models. The regularizing factors of these selectors are estimated via cross-validation procedures.

In [49], filter methods such as CC analysis, ITSS and Lipschitz quotients analysis are combined with either LASSO and plant experts' knowledge, halving the original input set of an SS of a refinery process and outperforming the model trained with all the candidate inputs.

Other cases of hybrid approaches are reported in [5].

6. Summary and Conclusions

Given the number of classes of input selection strategies, some key factors must be taken into account when creating the model. First of all, the designer needs to understand if the chosen algorithm is able to detect nonlinear relationships, which is a common trait when dealing with industrial processes. The number of available samples with respect to the number of inputs to be chosen could give a hint as well whether a filter, wrapper or embedded approach is preferable to avoid overfitting or poor generalization properties. The expected computational demand represents another incisive factor to be considered as well by the designer. Taking these considerations into account, Table 1 carries a classification of the methods cited so far, divided by classes and showing the benefits and drawbacks of each, with the aim to provide the designer with a guidance of what the most suitable choice could be for the case in exam.

Table 1. Classification of the methods, considering pros and cons. Given n number of samples and p number of inputs.

Class	Type	Pros	Cons

FE	**Linear**	**Nonlinear**		
_	PCA, SPCA, PPCA.	NLPCA, KPCA, ICA, MDS, PCoA, Isomap, LLE, LE, SOM.	Reduced computational demand.	Unsupervised. Final projections do not have any physical meaning and all measurement sensors are still needed.
FS	**Linear**	**NonLinear**		
Filter	CC analysis. (Pearson, Spearman)	Ensemble CC an. or MI an. (MIFS, MIFS-U, NMIFS, DMIFS, ITSS), Lipschitz coeff.	Simplest, fastest, model-independent. Good when $n < p$.	Inputs are considered individually. Dependencies and interactions are disregarded.
Wrapper	**Deterministic** FS, SFFS, BE, SBFS.	**Random** Heuristic search (GA, ACO, SA).	Evaluation on the final model gives very good results.	Model-dependent. Most computationally and time expensive. Models obtained can suffer overfitting.
Embedded	RFE, RFEST, EnRFE, Sensitivity analysis, Evolutionary ANNs, LASSO, LASSO-MLP, Elastic net.		Best methods when $n > p$.	Model-dependent. Computationally expensive. High overfitting when $n < p$.
Hybrid	Every possible combination of methods from different classes.		Merge best results from the most performing methods for the case in exam.	Different tests have to be done, methods have to be combined with a criterion. This can make them time consuming.

Wrappers and embedded algorithms are typically preferred where the number of candidate inputs is relatively smaller then the number of samples. Under this circumstance they both tend to give the best results even if they are time and computationally demanding. Otherwise, the final model will suffer overfitting, being the two methods model-dependent. As opposed to such model-dependent methods, filter approaches offer a faster and model-independent alternative. They perform an estimation of the input variable importance, avoiding this way the risk of overfitting. The input variables pruning ensures a reduction of the computational burden required for the model identification and validation steps. In some cases, such ranking can anyway be too inaccurate and an importance-wise greedy selection of the candidate inputs tends to ignore redundancies. For this reason, they work best as a first step of hybrid approaches. They represent the best choice if the number of candidate inputs is relatively greater then the number of samples.

Moreover, references were summarized into the next two tables: in Table 2, references explaining theory and procedures are classified for each input selection method; Table 3 collects references with real case studies application of each method.

Soft sensors concern several fields of study and research, from machine learning, mathematics and statistics. The choice of input variables has an utter impact in the model development and its final performance. In the case of empirical data-driven models, the difficulty of the task can be lightened by the a priori knowledge given by plant experts, if available. Most of the time, however, this is not possible, and a black-box investigation among variables is needed. Despite the variety of the existing methods, none of them

provides a general solution equally satisfactory for any case, since in the case of real applications, each approach tends to give different incoherent results. This means that different tests must be performed by the designer in an effort to find the most suitable subset for the application, making the input selection step time and computationally consuming. For this reason, the problem of variable selection is highly demanded, making it a topic that still needs to be researched.

Table 2. Classification of references of theory and procedure of the main input selection methods.

Methods References Table			
FE		PCA	[105–110,117]
		MDS	[118,119]
		PCoA	[120]
		Isomap	[121–123]
		LLE	[124,125]
		LE	[126]
		SOM	[127]
FS	*Filter*	CC	[1,136,138–141]
		Univariate MI	[129]
		Multivariate MI	[149,150,152–156,159]
		ITSS	
		Lipschitz quot.	[160]
			[161]
	Wrapper		[45]
		FS, BE	[130]
		SFFS, SFBS	[131]
		Random	[173–175]
		ACO based	[132–134]
	Embedded	RFE	[129,180,181]
		Sensitivity analysis	[182–184]
		EANN	[187]
		LASSO	[188,189,192]
Semi-supervised			[46]

Table 3. Classification of references of application of the methods on a real case study.

Real Case Applications References Table		
Plant experts' knowledge		[7–10,22,26,64]
FE	PCA	[11,19,28,34,36,37,56,58,89,111,112]
	Distributed PCA	[31]
	Kernel PCA	[113–115]
		[32,84]
	PLS	[16]
	Discriminant anal.	

FS	*Filter*	CC	[6,12,23,30,49,137]
		Univariate MI	[143]
		Multivariate MI	[75,77,90–92,144–146,156–158]
		ITSS	
		Lipschitz quot.	[13,49]
			[49,162]
	Wrapper	FS, BE	[14,99,165]
		SFFS, SFBS	[164]
		Random	[166–168,170,176,177]
	Embedded	RFE	[178,179]
		Sensitivity analysis	[185]
		LASSO	[49,162]
Hybrid			[29,49,83,193–197]

References

1. Fortuna, L.; Graziani, S.; Rizzo, A.; Xibilia, M.G. *Soft Sensors for Monitoring and Control of Industrial Processes*, 1st ed.; Springer: London, UK, 2007.
2. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814, doi:10.1016/j.compchemeng.2008.12.012.
3. Wold, S. Chemometrics; what do we mean with it, and what do we want from it? *Chem. Intell. Lab. Syst.* **1995**, *30*, 109–115, doi:10.1016/0169-7439(95)00042-9.
4. Otto, M. *Chemometrics: Statistics and Computer Application in Analytical Chemistry*, 3d ed.; Wiley: Hoboken, NJ, USA, 2016.
5. Souza, F.A.A.; Araújo, R.; Mendes, J. Review of Soft Sensors Methods for Regression Applications. *Chem. Intell. Lab. Syst.* **2016**, *152*, 69–79, doi:10.1016/j.chemolab.2015.12.011.
6. Fortuna, L.; Giannone, P.; Graziani, S.; Xibilia, M.G. Virtual Instruments Based on Stacked Neural Networks to Improve Product Quality Monitoring in a Refinery. *IEEE Trans. Instrum. Meas.* **2007**, *56*, 95–101, doi:10.1109/TIM.2006.887331.
7. Fortuna, L.; Graziani, S.; Xibilia, M.G. Virtual instruments in refineries. *IEEE Instrum. Meas. Mag.* **2005**, *8*, 26–34, doi:10.1109/MIM.2005.1518619.
8. Fortuna, L.; Graziani, S.; Xibilia, M.G. Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Eng. Pract.* **2005**, *13*, 499–508, doi:10.1016/j.conengprac.2004.04.013.
9. Fortuna, L.; Graziani, S.; Xibilia, M.G.; Barbalace, N. Fuzzy activated neural models for product quality monitoring in refineries. *IFAC Proc. Vol.* **2005**, *38*, 159–164, doi:10.3182/20050703-6-CZ-1902.01602.
10. Graziani, S.; Xibilia, M.G. Design of a Soft Sensor for an Industrial Plant with Unknown Delay by Using Deep Learning. In Proceedings of the 2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Auckland, New Zealand, 20–23 May 2019; pp. 1–6, doi:10.1109/I2MTC.2019.8827074.
11. Pani, A.K.; Amin, K.G.; Mohanta, H.K. Soft sensing of product quality in the debutanizer column with principal component analysis and feed-forward artificial neural network. *Alex. Eng. J.* **2016**, *55*, 1667–1674, doi:10.1016/j.aej.2016.02.016.
12. Graziani, S.; Xibilia, M.G. Deep Structures for a Reformer Unit Soft Sensor. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics

(INDIN), Porto, Portugal, 18–20 July 2018; pp. 927–932, doi:10.1109/INDIN.2018.8471942.
13. Stanišic´, D.; Jorgovanovic´, N.; Popov, N.; Congradac, V. Soft sensor for real-time cement fineness estimation.˘ *ISA Trans.* **2015**, *55*, 250–259, doi:10.1016/j.isatra.2014.09.019.
14. Seraj, M.; Aliyari Shoorehdeli, M. Data-driven predictor and soft-sensor models of a cement grate cooler based on neural network and effective dynamics. In Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 2–4 May 2017; doi:10.1109/IranianCEE.2017.7985134.
15. Bhavani, N.P.G.; Sujatha, K.; Kumaresan, M.; Ponmagal, R.S.; Reddy, S.B. Soft sensor for temperature measurement in gas turbine power plant. *Int. J. App. Eng. Res.* **2014**, *9*, 21305–21316.
16. Sujatha, K.; Bhavani, N.P.G.; Cao S.-Q.; Kumar, K.S.R. Soft Sensor for Flame Temperature Measurement and IoT based Monitoring in Power Plants. *Mater. Proc.* **2018**, *5*, 10755–10762, doi:10.1016/j.matpr.2017.12.359.
17. Amazouz, M.; Champagne, M.; Platon, R. Soft sensor application in the pulp and paper industry: Assessment study. Available online: https://www.researchgate.net/publication/280256929 (accessed on 3 June 2020). doi:10.13140/RG.2.1.2312.9442.
18. Runkler, T.; Gerstorfer, E.; Schlang, M.; Jj, E.; Villforth, K. Data Compression and Soft Sensors in the Pulp and Paper Industry. 2015. Available online: https://www.researchgate.net/publication/266507240_Data_compression_and_soft_sensors_in_the_pulp_and_paper_industry (accessed on 3 June 2020).
19. Osorio, D.; Pérez-Correa, J.; Agosin, E.; Cabrera, M. Soft-sensor for on-line estimation of ethanol concentrations in wine stills. *J. Food Eng.* **2008**, *87*, 571–577, doi:10.1016/j.jfoodeng.2008.01.011.
20. Rizzo, A. Soft sensors and artificial intelligence for nuclear fusion experiments. In Proceedings of the 15th IEEE Mediterranean Electrotechnical Conference, Valletta, Malta, 26–28 April 2010; pp. 1068–1072, doi:10.1109/MELCON.2010.5476042.
21. Fortuna, L.; Fradkov, A.; Frasca, M. *From Physics to Control Through an Emergent View*; From World Scientific Series on Nonlinear Science Series B, Volume 15; World Scientific: London, UK, 2010; doi:10.1142/7790.
22. Fortuna, L.; Graziani, S.; Xibilia, M.G.; Napoli, G. Virtual Instruments for the what-if analysis of a process for pollution minimization in an industrial application. In Proceedings of the 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 28–30 June 2006; pp. 1–4, doi:10.1109/MED.2006.328755.
23. Gonzaga, J.C.B.; Meleiro, L.A.C.; Kiang, C.; Filho, R.M. Ann-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Comput. Chem. Eng.* **2009**, *33*, 43–49, doi:10.1016/j.compchemeng.2008.05.019.
24. Frauendorfer, E.; Hergeth, W.-D. Soft Sensor Applications in Industrial Vinylacetate-ethylene (VAE) Polymerization Processes. *Macromol. React. Eng.* **2017**, *11*, doi:10.1002/mren.201700008.
25. Zhu, C.-H.; Zhang, J. Developing Soft Sensors for Polymer Melt Index in an Industrial Polymerization Process Using Deep Belief Networks. *Int. J. Aut. Comput.* **2020**, *117*, 44–54, doi:10.1007/s11633-019-1203-x.
26. Graziani, S.; Xibilia, M.G. A deep learning based soft sensor for a sour water stripping plant. In Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Turin, Italy, 22–25 May 2017; pp. 1–6, doi:10.1109/i2mtc.2017.7969924.

27. Pisa, I.; Santin, I.; Vicario, J.L.; Morell, A.; Vilanova, R. ANN-Based Soft Sensor to Predict Effluent Violations in Wastewater Treatment Plants. *Sensors* **2019**, *19*, 1280, doi:10.3390/s19061280.
28. Choi, D.J.; Park, H. A hybrid artificial neural network as a software sensor for optimal control of a wastewater treatment process. *Wat. Res.* **2001**, *35*, 3959–3967, doi:10.1016/S0043-1354(01)00134-8.
29. Bowden, G.J.; Dandy, G.C.; Maier, H.R. Input determination for neural network models in water resources applications. Part 1—Background and methodology. *J. Hydrol.* **2005**, *301*, 75–92, doi:10.1016/j.jhydrol.2004.06.021. 30. Andò, B.; Graziani, S.; Xibilia, M.G. Low-order Nonlinear Finite-Impulse Response Soft Sensors for Ionic Electroactive Actuators based on Deep Learning. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 1637–1646, doi:10.1109/TIM.2018.2884450.
31. Bao, Y.; Zhu, Y.; Du, W.; Zhong, W.; Quian, F. A distributed PCA-TSS based soft sensor for raw meal finesses in VRM system. *Control Eng. Pract.* **2019**, *90*, 38–49, doi:10.1016/j.conengprac.2019.06.009.
32. He, K.; Quian, F.; Cheng, H.; Du, W. A novel adaptive algorithm with near-infrared spectroscopy and its applications in online gasoline blending processes. *Chem. Intell. Lab. Syst.* **2015**, *140*, 117–125, doi:10.1016/j.chemolab.2014.11.006.
33. Kang, J.; Shao, Z.; Chen, X.; Gu, X.; Feng, L. Fast and reliable computational strategy for developing a rigorous model-driven soft sensor of dynamic molecular weight distribution. *J. Process Control* **2017**, *56*, 79–99, doi:10.1016/j.jprocont.2017.05.006.
34. Mandal, S.; Santhi, B.; Sridar, S.; Vinolia, K.; Swaminathan, P. Sensor fault detection in nuclear power plant using statistical methods. *Nucl. Eng. Des.* **2017**, *324*, 103–110, doi:10.1016/j.nucengdes.2017.08.028.
35. Marinkovic, Z.; Atanaskovic, A.; Xibilia, M.G.; Pace, C.; Latino, M.; Donato, N. A neural network approach for safety monitoring applications. In Proceedings of the 2016 IEEE Sensors Applications Symposium (SAS), Catania, Italy, 20–22 April 2016; pp. 297–301, doi:10.1109/SAS.2016.7479862.
36. Zamprogna, E.; Barolo, M.; Seborg, D.E. Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis. *J. Process Control* **2005**, *15*, 39–52, doi:10.1016/j.jprocont.2004.04.006.
37. Liu, Y.; Huang, D.; Li, Y.; Zhu, X. Development of a novel self-validation soft sensor. *Korean J. Chem. Eng.* **2012**, *29*, 1135–1143, doi:10.1007/s11814-011-0289-9.
38. Yao, L.; Ge, Z. Moving window adaptive soft sensor for state shifting process based on weighted supervised latent factor analysis. *Control Eng. Pract.* **2017**, *61*, 72–80, doi:10.1016/j.conengprac.2017.02.002.
39. Nassour, J.; Ghadiya, V.; Hugel, V.; Hamker, F.H. Design of new Sensory Soft Hand: Combining air-pump actuation with superimposed curvature and pressure sensors. In Proceedings of the IEEE International Conference on Soft Robotics (RoboSoft), Livorno, Italy, 24–28 April 2018; pp. 164–169, doi:10.1109/ROBOSOFT.2018.8404914.
40. Vogt, D.; Park, Y.; Wood, R.J. A soft multi-axis force sensor. In Proceedings of the 2012 IEEE, Taipei, Taiwan, 28–31 October 2012; pp. 1–4, doi:10.1109/ICSENS.2012.6411573.
41. Xu, S.; Vogt, D.M.; Hsu, W.-H.; Osborne , J.; Walsh, T.; Foster, J.R.; Sullivan, S.K.; Smith, V.C.; Rousing, A.W.; Goldfield, E.C.; et al. Biocompatible Soft Fluidic Strain and Force Sensors for Wearable Devices. *Adv. Funct. Mater.* **2019**, *29*, doi:10.1002/adfm.201807058.
42. May, R.J.; Dandy, G.C.; Mayer, H.R. Review of Input Variable Selection Methods for Artificial Neural Networks. In *Artificial Neural Networks—Methodological Advances*

and Biomedical Applications; Suzuki, K., Ed.; InTechOpen: London, UK, 2011; pp. 19–44, doi:10.5772/16004.

43. Hira, Z.; Gillies, D. A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Adv. Bioinf.* **2015**, *2015*, 1–13, doi:10.1155/2015/198363.
44. Vergara, J.R.; Estévez, P.A. A review of feature selection methods based on mutual information. *Neural Comput. Appl.* **2014**, *24*, 175–186, doi:10.1007/s00521-013-1368-0.
45. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324, doi:10.1016/S0004-3702(97)00043-X.
46. Sheikhpour, R.; Sarram, M.A.; Gharaghani, S.; Chahooki, M.A.Z. A Survey on semi-supervised feature selection methods. *Patton Rec.* **2017**, *64*, 141–158, doi:10.1016/j.patcog.2016.11.003.
47. ABB, Human in the Loop. Abb Review 2007, January 2007. Available online: https://library.e.abb.com/ public/b9f582f7087d8a27c125728b0047ce18/Review_1_2007_72dpi.pdf (accessed on 3 June 2020).
48. Cimini, C.; Pirola, F.; Pinto, R.; Cavalieri, S. A human-in-the-loop manufacturing control architecture for the next generation of production systems. *J. Manuf. Syst.* **2020**, *54*, 258–271, doi:10.1016/j.jmsy.2020.01.002.
49. Curreri, F.; Graziani, S.; Xibilia, M.G. Input selection methods for data-driven Soft Sensors design: Application to an industrial process. *Inf. Sci.* **2020**, doi:10.1016/j.ins.2020.05.028.
50. Bishop, C.M. *Pattern Recognition and Machine Learning*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2006. 51. Ljung, L. *System Identification: Theory for the User*, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1999.
52. Boyer, S.A. *SCADA: Supervisory Control and Data Acquisition*, 4th ed.; ISA—Internation Society of Automation: Pittsburgh, PA, USA, 2010.
53. D'Andrea, R. ; Dullerud, G.E. Distributed control design for spatially interconnected systems. *IEEE Trans. Autom. Control* **2003**, *48*, 1478–1495, doi:10.1109/tac.2003.816954.
54. Di Bella, A.; Fortuna, L.; Graziani, S.; Napoli, G.; Xibilia, M.G. A comparative analysis of the influence of methods for outliers detection on the performance of data driven models. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference, Warsaw, Poland, 1–3 May 2007; doi:10.1109/IMTC.2007.379222.
55. Ge, Z. Active probabilistic sample selection for intelligent soft sensing of industrial processes. *Chem. Intell. Lab. Syst.* **2016**, *152*, 181–189, doi:10.1016/j.chemolab.2016.01.003.
56. Lin, B.; Knudsen, J.K.H.; Jorgensen, S.B. A systematic approach for soft sensor development. *Comput. Chem. Eng.* **2007**, *31*, 419–425, doi:10.1016/j.compchemeng.2006.05.030.
57. Xie, L.; Yang, H.; Huang, B. Fir model identification of multirate processes with random delays using em algorithm. *AIChE J.* **2013**, *59*, 4124–4132, doi:10.1002/aic.14147.
58. Kadlec, P.; Gabrys, B. Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE J.* **2011**, *57*, 1288–1301, doi:10.1002/aic.12346.
59. Lu, L.; Yang, Y.; Gao, F.; Wang, F. Multirate dynamic inferential modeling for multivariable processes. *Chem. Eng. Sci.* **2004**, *59*, 855–864, doi:10.1016/j.ces.2003.12.003.
60. Wu, Y.; Luo, X. A novel calibration approach of soft sensor based on multirate data fusion technology. *J. Proc Control* **2010**, *20*, 1252–1260, doi:10.1016/j.jprocont.2010.09.003.

61. Pearson, R.K. Outliers in process modeling and identification. *IEEE Trans. Control Syst. Tech.* **2002**, *10*, 55–63, doi:10.1109/87.974338.
62. Piltan, F.; TayebiHaghighi, S.; Sulaiman, N.B. Comparative study between ARX and ARMAX system identification. *Int. J. Intell. Control Appl.* **2017**, *9*, 25–34, doi:10.5815/ijisa.2017.02.04.
63. Sjöberg, J.; Zhang, Q.; Ljung, L.; Benveniste, A.; Delyon, B.; Glorennec, P.-Y.; Hjalmarsson, H.; Juditsky, A. *Nonlinear Black-Box Modeling in System Identification: A Unified Overview*; Linköping University: Linköping, Sweden, 1995; Volume 31, pp. 1691–1724, doi:10.1016/0005-1098(95)00120-8.
64. Fortuna, L.; Graziani, S.; Xibilia, M.G. Comparison of soft-sensor design methods for industrial plants using small data sets. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 2444–2451, doi:10.1109/TIM.2009.2016386.
65. Wei, J.; Guo, L.; Xu, X.; Yan, G. Soft sensor modeling of mill level based on convolutional neural network. In Proceedings of the 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 23–25 May 2015; doi:10.1109/CCDC.2015.7162762.
66. Wang, K.; Shang, C.; Liu, L.; Jiang, Y.; Huang, D.; Yang, F. Dynamic Soft Sensor Development Based on Convolutional Neural Networks. *Ind. Eng. Chem. Res.* **2019**, *58*, 11521–11531, doi:10.1021/acs.iecr.9b02513.
67. Wang, X. Data Preprocessing for Soft Sensor Using Generative Adversarial Networks. In Proceedings of the International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; doi:10.1109/ICARCV.2018.8581249.
68. Hong, Y.; Hwang, U.; Yoo, J.; Yoon, S. How Generative Adversarial Networks and Their Variants Work: An Overview. *arXiv* **2019**, arXiv:1711.05914.
69. Wang, J.-S.; Han, S.; Yang, Y. RBF Neural Network Soft-Sensor Model of Electroslag Remelting Process Optimized by Artificial Fish Swarm Optimization Algorithm. *Adv. Mech. Eng.* **2015**, *6*, doi:10.1155/2014/318195.
70. Gao, M.-J.; Tian, J.-W.; Li, K. The study of soft sensor modeling method based on wavelet neural network for sewage treatment. In Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, 2–4 November 2007; pp. 721–726, doi:10.1109/ICWAPR.2007.4420763.
71. Wei, Y.; Jiang, Y.; Yang, F.; Huang, D. Three-Stage Decomposition Modeling for Quality of Gas-Phase Polyethylene Process Based on Adaptive Hinging Hyperplanes and Impulse Response Template. *Ind. Eng. Chem. Res.* **2013**, *52*, 5747–5756, doi:10.1021/ie303370x.
72. Lu, M.; Kang, Y.; Han, X.; Yan, G. Soft sensor modeling of mill level based on Deep Belief Network. In Proceedings of the 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, China, 31 May–2 June 2014; pp. 189–193, doi:10.1109/CCDC.2014.6852142.
73. Liu, R.; Rong, Z.; Jiang, B.; Pang, Z.; Tang, C. Soft Sensor of 4-CBA Concentration Using Deep Belief Networks with Continuous Restricted Boltzmann Machine. In Proceedings of the 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Nanjing, China, 23–25 November 2018; pp. 421–424, doi:10.1109/CCIS.2018.8691166.
74. Wang, X.; Liu, H. Soft sensor based on stacked auto-encoder deep neural network for air preheater rotor deformation prediction. *Adv. Eng. Inf.* **2018**, *36*, 112–119, doi:10.1016/j.aei.2018.03.003.

75. Li, D.; Li, Z.; Sun, K. Development of a Novel Soft Sensor with Long Short-Term Memory Network and Normalized Mutual Information Feature Selection. *Math. Probl. Eng.* **2020**, doi:10.1155/2020/7617010.
76. Chitralekha, S.B.; Shah, S. Support Vector Regression for soft sensor design of nonlinear processes. In Proceedings of the 18th Mediterranean Conference on Control and Automation, Marrakech, Morocco, 23–25 June 2010; pp. 569–574, doi:10.1109/MED.2010.5547730.
77. Grbic´, R.; Sliskovic, D.; Kadlec, P. Adaptive soft sensor for online prediction and process monitoring based on mixture of Gaussian process models. *Comput. Chem. Eng.* **2013**, *58*, 84–97, doi:10.1016/j.compchemeng.2013.06.014.
78. Shao, W.; Ge, Z.; Song, Z.; Wang, K. Nonlinear Industrial soft sensor development based on semi-supervised probabilistic mixture of extreme learning machines. *Control Eng. Pract.* **2019**, *91*, doi:10.1016/j.conengprac.2019.07.016.
79. Mendes, J.; Souza, F.; Araújo, R.; Gonçalves, N. Genetic fuzzy system for data-driven soft sensors. *Appl. Soft Comput.* **2012**, *12*, 3237–3245, doi:10.1016/j.asoc.2012.05.009.
80. Mendes, J.; Pinto, S.; Araújo, R.; Souza, F. Evolutionary fuzzy models for nonlinear identification.
In Proceedings of the IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), Krakow, Poland, 17–21 September 2012; pp. 1–8, doi:10.1109/ETFA.2012.6489621.
81. Sjöberg, J.; Hjalmarsson, H.; Ljung, L. Neural networks in system identification. *IFAC Proc. Vol.* **1994**, *27*, 359–382, doi:10.1016/S1474-6670(17)47737-8.
82. Juditsky, A.; Hjalmarsson, H.; Benveniste, A.; Delyon, B.; Ljung, L.; Sjoberg, J.; Zhang, Q. Nonlinear black-box models in system identification: Mathematical foundations. *Automatica* **1995**, *31*, 1725–1750, doi:10.1016/0005-1098(95)00119-1.
83. Han, M.; Zhang, R.; Xu, M. Multivariate chaotic time series prediction based on ELM-PLSR and hybrid variable selection algorithm. *Neural Proc. Lett.* **2017**, *46*, 705–717, doi:10.1007/s11063-017-9616-4.
84. Liu, Y.; Pan, Y.; Huang, D. Development of a novel adaptive soft sensor using variational Bayesian PLS with accounting for online identification of key variables. *Ind. Eng. Chem. Res.* **2015**, *54*, 338–350, doi:10.1021/ie503807e.
85. Liu, Z.; Ge, Z.; Chen, G.; Song, Z. Adaptive soft sensors for quality prediction under the framework of Bayesian network. *Control Eng. Pract.* **2018**, *72*, 19–29, doi:10.1016/j.conengprac.2017.10.018.
86. Graziani, S.; Xibilia, M.G. *Deep Learning for Soft Sensor Design, in Development and Analysis of Deep Learning Architectures*; Springer: Basel, Switzerland, 2020, doi:10.1007/978-3-030-31764-5_2.
87. Shoorehdeli, M.A.; Teshnehlab, M.; Sedigh, A.K. Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter. *Fuzzy Sets Syst.* **2009**, *160*, 922–948, doi:10.1016/j.fss.2008.09.011.
88. Soares, S.; Araújio, R.; Sousa, P.; Souza, F. Design and application of soft sensors using ensemble methods. In Proceedings of the IEEE International Conference on Emerging Technologies & Factory Automation, ETFA2011, Toulouse, France, 5–9 September 2011; pp. 1–8, doi:10.1109/ETFA.2011.6059061.
89. Shakil, M.; Elshafei, M.; Habib, M.A.; Maleki, F.A. Soft sensor for NOx and O2 using dynamic neural networks. *Comput. Elect. Eng.* **2009**, *35*, 578–586, doi:10.1016/j.compeleceng.2008.08.007.

90. Ludwig, O.; Nunes, U.; Araújo, R.; Schnitman, L.; Lepikson, H.A. Applications of information theory, genetic algorithms, and neural models to predict oil flow. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 2870–2885, doi:10.1016/j.cnsns.2008.12.011.
91. Souza, F.; Araújo, R. Variable and time-lag selection using empirical data. In Proceedings of the IEEE International Conference on Emerging Technologies & Factory Automation, ETFA2011, Toulouse, France, 5–9 September 2011; pp. 1–8, doi:10.1109/ETFA.2011.6059083.
92. Souza, F.; Santos, P.; Araújio, R. Variable and delay selection using neural networks and mutual information for data-driven soft sensors. In Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Bilbao, Spain, 13–16 September 2010; pp. 1–8, doi:10.1109/ETFA.2010.5641329.
93. Lou, H.; Su, H.; Xie, L.; Gu, Y.; Rong, G. Inferential Model for Industrial Polypropylene Melt Index Prediction with Embedded Priori Knowledge and Delay Estimation. *Ind. Eng. Chem. Res.* **2012**, *51*, 8510–8525, doi:10.1021/ie202901v.
94. Akaike, H. Information Theory and an Extension of the Maximum Likelihood Principle. In *Selected Papers of Hirotugu Akaike*; Springer: New York, NY, USA, 1998; pp. 199–213, doi:10.1007/978-1-4612-1694-0_15.
95. Akaike, H. A new look at the statistical model identification. *IEEE Trans. Autom. Control* **1974**, *19*, 716–723, doi:10.1109/TAC.1974.1100705.
96. Rissanen, J. Modeling by shortest data description. *Automatica* **1974**, *14*, 465–658, doi:10.1016/0005-1098(78)90005-5. 97. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464, doi:10.1214/aos/1176344136.
98. Mallows, C.L. Some Comments on C_P. *Technometrics* **1973**, *15*, 661–675, doi:10.2307/1267380.
99. Gabriel, D.; Matias, T.; Pereira, J.C.; Araújo, R. Predicting gas emissions in a cement kiln plant using hard and soft modeling strategies. In Proceedings of the IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, Italy, 10–13 September 2013; pp. 1–8, doi:10.1109/ETFA.2013.6648036.
100. Judd, J.S. *Neural Network Design and the Complexity of Learning*; The MIT Press: Cambridge, MA, USA, 1990.
101. Geman, S.; Bienenstock, E.; Doursat, R. Neural Networks and the Bias/Variance Dilemma. *Neural Comput.* **1992**, *4*, 1–58, doi:10.1162/neco.1992.4.1.1.
102. Bellman, R. *Adaptive Control Processes: A Guided Tour*; Princeton University Press: New Jersey, NJ, USA, 1961. 103. Scott, D.W. *Multivariate Density Estimation: Theory, Practice and Visualisation*; John Wiley and Sons: New York, NY, USA, 1992.
104. Jain, A.K.; Duin, R.P.; Mao, J. Statistical pattern recognition: A review. *IEEE Trans. Patton Anal. Mach. Intell.* **2000**, *22*, 4–37, doi:10.1109/34.824819.
105. Joliffe, I.T. *Principal Component Analysis*; Springer: Berlin/Heidelberg, Germany, 2002.
106. Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* **1991**, *37*, 233–243, doi:10.1002/aic.690370209.
107. Schölkopf, B. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Comput.* **1998**, *10*, 1299–1319, doi:10.1162/089976698300017467.
108. Bair, E.; Hastie, T; Paul, D.; Tibshirani, R. Prediction by Supervised Principal Components. *J. Am. Stat. Assess.* **2006**, *101*, 119–137, doi:10.1198/016214505000000628.
109. Comon, P. Independent component analysis, A new concept? *Signal Proc.* **1994**, *36*, 287–314, doi:10.1016/0165-1684(94)90029-9.

110. Tipping, M.E.; Bishop, C.M. Probabilistic Principal Component Analysis. *J. R. Stat. Soc. Ser. B* **1999**, *61*, 611–622, doi:10.1111/1467-9868.00196.
111. Eshghi, P. Dimensionality choice in principal components analysis via cross-validatory methods. *Chem. Intell. Lab. Syst.* **2014**, *130*, 6–13, doi:10.1016/j.chemolab.2013.09.004.
112. Hastie, T.; Tibshirani, R.; Eisen, M.B.; Alizadeh, A.; Levy, R.; Staudt, L.; Chan, W.C.; Botstein, D.; Brown, P.
'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biol.* **2000**, *1*, 1–21, doi:10.1186/gb-2000-1-2-research0003.
113. Liu, Z.; Chen, D.; Bensmail, H. Gene expression data classification with kernel principal component analysis. *BioMed Res. Int.* **2005**, *37*, 155–159, doi:10.1155%2FJBB.2005.155.
114. Reverter, F.; Vegas, E.; Oller, J.M. Kernel-PCA data integration with enhanced interpretability. *BMC Syst. Biol.* **2014**, *8*, doi:10.1186/1752-0509-8-S2-S6.
115. Yao, M.; Wang, H. On-line monitoring of batch processes using generalized additive kernel principal component analysis. *J. Proc. Control* **2015**, *28*, 56–72, doi:10.1016/j.jprocont.2015.02.007.
116. Cao, L.J.; Chua, K.S.; Chong, W.K.; Lee, H.P.; Gu, Q.M. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *IEEE Trans. Patton Anal. Mach. Intell.* **2003**, *55*, 321–336, doi:10.1016/S0925-2312(03)00433-8.
117. Li, T.; Su, Y.; Yi, J.; Yao, L.; Xu, M. Original feature selection in soft-sensor modeling process based on ICA_FNN. *Chin. J. Sci. Instrum.* **2013**, *4*, 736–742.
118. Torgerson, W.S. Multidimensional scaling: I. Theory and method. *Psychometrika* **1952**, *17*, 401–159, doi:10.1007/BF02288916.
119. Borg, I.; Groenen, P.J.F. *Modern Multidimensional Scaling: Theory and Applications*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2005.
120. Gower, J.C. Principal Coordinates Analysis. *Encycl. Biostat.* **2005**, doi:10.1002/0470011815.b2a13070.
121. Tenenbaum, J.B.; De Silva, V.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323, doi:10.1126/science.290.5500.2319.
122. Balasubramanian, M.; Schwartz, E.L. The isomap algorithm and topological stability. *Science* **2002**, *295*, 7, doi:10.1126/science.295.5552.7a.
123. Orsenigo, C.; Vercellis, C. An effective double-bounded tree-connected Isomap algorithm for microarray data classification. *Patton Rec. Lett.* **2012**, *33*, 9–13, doi:10.1016/j.patrec.2011.09.016.
124. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323–2326.
125. Shi C.; Chen, L. Feature dimension reduction for microarray data analysis using locally linear embedding. In Proceedings of the 3rd Asia-Pacific Bioinformatics Conference, APBC '05, Singapore, 17–21 January 2005; pp. 211–217, doi:10.1142/9781860947322_0021.
126. Belkin, M.; Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* **2003**, *15*, 1373–1396, doi:10.1162/089976603321780317.
127. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **1982**, *43*, 56–69, doi:10.1007/BF00337288.
128. Reitermanová, Z. Information Theory Methods for Feature Selection, 2010. Available online: https://pdfs.

semanticscholar.org/ad7c/9cbb5411a4ff10cec3c9ac5ddc18f1f60979.pdf (accessed on 3 June 2020).

129. Guyon, I. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
130. Aha, D.W.; Bankert, R.L. A Comparative Evaluation of Sequential Feature Selection Algorithms. In *Learning from Data*; Lecture Notes in Statistics; Springer: New York, NY, USA, 1996; Volume 112, pp. 199–206, doi:10.1007/978-1-4612-2404-4_19.
131. Somol, P.; Novovicova, J.; Pudil, P. Efficient Feature Subset Selection and Subset Size Optimization. *Patton Rec. Recent Adv.* **2010**, *56*, doi:10.5772/9356.
132. Izrailev, S.; Agrafiotis, D.K. Variable selection for QSAR by artificial ant colony systems. *SAR QSAR Environ. Res.* **2002**, *13*, 417–423, doi:10.1080/10629360290014296.
133. Marcoulides, G.A.; Drezner, Z. Model specification searches using ant colony optimization algorithms. *Struct. Eq. Model.* **2003**, *10*, 154–164, doi:10.1207/S15328007SEM1001_8.
134. Shen, Q.; Jiang, J.-H.; Tao, J.-C.; Shen, G.-L.; Yu, R.-Q. Modified ant colony optimization algorithm for variable selection in QSAR modeling: QSAR studies of cyclooxygenase inhibitors. *J. Chem. Inf. Model.* **2005**, *45*, 1024–1029, doi:10.1021/ci049610z.
135. Tang, J.; Alelyani, S.; Liu, H. Feature selection for classification: A review. *Data Class. Algorithms Appl.* **2014**, 37–64, doi:10.1201/b17320.
136. Chen, P.Y.; Popovich, P.M. *Correlation: Parametric and Nonparametric Measures*; Sage Publications: Thousand Oaks, CA, USA, 2002.
137. Delgado, M.R.; Nagai, E.Y.; Arruda, L.V.R. A neuro-coevolutionary genetic fuzzy system to design soft sensors. *Soft Comput.* **2009**, *13*, 481–495, doi:10.1007/s00500-008-0363-3.
138. Deebani, W.; Kachouie, N.N. Ensemble Correlation Coefficient. In Proceedings of the International Symposium on Artificial Intelligence and Mathematics, ISAIM 2018, Fort Lauderdale, FL, USA, 3–5 January 2018; Available online: https://dblp.org/rec/conf/isaim/DeebaniK18 (accessed on 3 June 2020).
139. Székely, G.J.; Rizzo, M.L.; Bakirov, N.K. Measuring and testing dependence by correlation of distances. *Ann. Stat.* **2007**, *35*, 2769–2794, doi:10.1214/009053607000000505.
140. Breiman, L.; Friedman, J.H. Estimating Optimal Transformations for Multiple Regression and Correlation. *J. Am. Stat. Assess.* **1985**, *80*, 580–598, doi:10.1080/01621459.1985.10478157.
141. Reshef, D.N.; Reshef, Y.A.; Mitzenmacher, M.M.; Sabeti, P.C. Equitability Analysis of the Maximal Information Coefficient, with Comparisons. *arXiv* **2013**, arXiv:1301.6314.
142. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; Wiley: Hoboken, NJ, USA, 1991.
143. Wang, X.; Han, M.; Wang, J. Applying input variables selection technique on input weighted support vector machine modeling for BOF endpoint prediction. *Eng. Appl. Artif. Intell.* **2010**, *23*, 1012–1018, doi:10.1016/j.engappai.2009.12.007.
144. Rossi, F.; Lendasse, A.; François, D.; Wertz, V.; Verleysen, M. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chem. Intell. Lab. Syst.* **2006**, *80*, 215–226, doi:10.1016/j.chemolab.2005.06.010.
145. François, D.; Rossi, F.; Wertz, V.; Verleysen, M. Resampling methods for parameter-free and robust feature selection with mutual information. *Neurocomp.* **2007**, *70*, 1276–1288, doi:10.1016/j.neucom.2006.11.019.

146. Xing, H.-J.; Hu, B.-G. Two-phase construction of multilayer perceptrons using information theory. *IEEE Trans. Neural Netw.* **2009**, *20*, 715–721, doi:10.1109/TNN.2008.2005604.
147. Doquire, G.; Verleysen, M. A comparison of multivariate mutual information estimators for feature selection. In Proceedings of the Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods, Algarve, Portugal, 6–8 February 2012; pp. 176–185, doi:10.5220/0003726101760185.
148. Frénay, B.; Doquire, G.; Verleysen, M. Is mutual information adequate for feature selection in regression? *Neural Netw.* **2013**, *48*, 1–7, doi:10.1016/j.neunet.2013.07.003.
149. Battiti, R. Using Mutual Information for Selecting Features in Supervised Neural Net Learning. *IEEE Trans. Neual Netw.* **1994**, *5*, 537–550, doi:10.1109/72.298224.
150. Kwak, N.; Choi, C.H. Input Feature Selection for Classification Problems. *IEEE Trans. Neural Netw.* **2002**, *13*, 143–159, doi:10.1109/72.977291.
151. Balagani, K.S.; Phoha, V.V. On the feature selection criterion based on an approximation of multidimensional mutual information. *IEEE Trans. Patton Anal. Mach. Intell.* **2010**, *32*, 1342–1343, doi:10.1109/TPAMI.2010.62. 152. Peng, H.; Long, F.; Ding, C. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. Patton Anal. Mach. Intell.* **2005**, *27*, 1226–1238, doi:10.1109/TPAMI.2005.159.
153. Che, J.; Yang, Y.; Li, L.; Bai, X.; Zhang, S.; Deng, C. Maximum relevance minimum common redundancy feature selection for nonlinear data. *Inf. Sci.* **2017**, *409–410*, 68–86, doi:10.1016/j.ins.2017.05.013.
154. Estévez, P.A.; Tesmer, M.; Perez, C.A.; Zurada, J.M. Normalized Mutual Information Feature Selection. *IEEE Trans. Neual Netw.* **2009**, *20*, 189–201, doi:10.1109/TNN.2008.2005601.
155. Liu, H.; Sun, J.; Liu, L.; Zhang, H. Feature selection with dynamic mutual information. *Patton Rec.* **2009**, *42*, 1330–1339, doi:10.1016/j.patcog.2008.10.028.
156. Sharma, A. Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1—A strategy for system predictor identification. *J. Hydrol.* **2000**, *239*, 232–239, doi:10.1016/S0022-1694(00)00346-2.
157. May, R.J.; Maier, H.R.; Dandy, G.C.; Fernando, T.M.K.G. Non-linear variable selection for artificial neural networks using partial mutual information. *Environ. Model. Softw* **2008**, *23*, 1312–1326, doi:10.1016/j.envsoft.2008.03.007.
158. Fernando, T.M.K.G.; Maier, H.R.; Dandy, G.C. Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. *J. Hydrol.* **2009**, *367*, 165–176, doi:10.1016/j.jhydrol.2008.10.019.
159. Liu, H.; Ditzler, G. A semi-parallel framework for greedy information-theoretic feature selection. *Inf. Sci.* **2019**, *492*, 13–28, doi:10.1016/j.ins.2019.03.075.
160. Sridhar, D.V.; Bartlett, E.B.; Seagrave, R.C. Information theoretic subset selection for neural network models.
Comput. Chem. Eng. **1998**, *22*, 613–626, doi:10.1016/S0098-1354(97)00227-5.
161. He, X.; Asada, H. A new method for identifying orders of input-output models for nonlinear dynamic systems. In Proceedings of the American Control Conference, San Francisco, CA, USA, 2–4 June 1993; doi:10.23919/ACC.1993.4793346.
162. Xibilia, M.G.; Gemelli, N.; Consolo, G. Input variables selection criteria for data-driven Soft Sensors design. In Proceedings of the IEEE International Conference on Networking, Sensing and Control, Calabria, Italy, 16–18 May 2017; doi:10.1109/ICNSC.2017.8000119.

163. Krakovska, O.; Christie, G.; Sixsmith, A.; Ester, M.; Moreno, S. Performance comparison of linear and non-linear feature selection methods for the analysis of large survey datasets. *PLoS ONE* **2019**, *14*, doi:10.1371/journal.pone.0213584.
164. Chu, Y.-H.; Lee, Y.-H.; Han, C. Improved Quality Estimation and Knowledge Extraction in a Batch Process by Bootstrapping-Based Generalized Variable Selection. *Ind. Eng. Chem. Res.* **2004**, *43*, 2680–2690, doi:10.1021/ie0341552.
165. Kaneko, H.; Funatsu, K. A new process variable and dynamics selection method based on a genetic algorithm-based wavelength selection method. *AIChE J.* **2012**, *58*, 1829–1840, doi:10.1002/aic.13814.
166. Chatterjee, S.; Bhattacherjee, A. Genetic algorithms for feature selection of image analysis-based quality monitoring model: An application to an iron mine. *Eng. Appl. Artif. Intell.* **2011**, *24*, 786–795, doi:10.1016/j.engappai.2010.11.009.
167. Arakawa, M.; Yamashita, Y.; Funatsu, K. Genetic algorithmbased wavelength selection method for spectral calibration. *J. Chem.* **2011**, *25*, 10–19, doi:10.1002/cem.1339.
168. Kaneko, H.; Funatsu, K. Nonlinear regression method with variable region selection and application to soft sensors. *Chem. Intell. Lab. Syst.* **2013**, *121*, 26–32, doi:10.1016/j.chemolab.2012.11.017.
169. Pierna J.A.F.; Abbas, O.; Baeten, V.; Dardenne, P. A backward variable selection method for PLS regression (BVSPLS). *Anal. Chim. Acta* **2009**, *642*, 89–93, doi:10.1016/j.aca.2008.12.002.
170. Liu, G.; Zhou, D.; Xu, H.; Mei, C. Model optimization of SVM for a fermentation soft sensor. *Exp. Syst. Appl.* **2010**, *37*, 2708–2713, doi:10.1016/j.eswa.2009.08.008.
171. Saeys, Y.; Inza, I.; Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics* **2007**, *23*, 2507–2517, doi:10.1093/bioinformatics/btm344.
172. Mehmood, T.; Liland, K.H.; Snipen, L.; Sæbø, S. A review of variable selection methods in Partial Least Squares Regression. *Chem. Intell. Lab. Syst.* **2012**, *118*, 62–69, doi:10.1016/j.chemolab.2012.07.010.
173. Centner, V.; Massart, D.L.; De Noord, O.; De Jong, S.; Vandeginste, B.G.M.; Sterna, C. Elimination of uninformative variables for multivariate calibration. *Anal. Chem.* **1996**, *68*, 3851–3858, doi:10.1021/ac960321m.
174. Li, H.-D.; Zeng, M.-M.; Tan, B.-B. ; Liang, Y.-Z.; Xu, Q.-S.; Cao, D.-S. Recipe for revealing informative metabolites based on model population analysis. *Metabolomics* **2010**, *6*, 353–361, doi:10.1007/s11306-010-0213-z.
175. Mehmood, T.; Martens, H.; Sæbø, S.; Warringer, J.; Snipen, L. A Partial least squares-based algorithm for parsimonious variable selection. *Algorithms Mol. Biol.* **2011**, *6*, doi:10.1186/1748-7188-6-27.
176. Cai, W.; Li, Y.; Shao, X. A variable selection method based on uninformative variable elimination for multivariate calibration of near-infrared spectra. *Chem. Intell. Lab. Syst.* **2008**, *90*, 188–194, doi:10.1016/j.chemolab.2007.10.001.
177. Hasegawa, K.; Miyashita, Y.; Funatsu, K. GA strategy for variable selection in QSAR studies: GA-based PLS analysis of calcium channel antagonists. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 306–310, doi:10.1021/ci960047x. 178. Liu, Q.; Sung, A.H.; Chen, Z.; Liu, J.; Huang, X.; Deng, Y. Feature selection and classification of MAQC-II breast cancer and multiple myeloma microarray gene expression data. *PLoS ONE* **2019**, *4*, doi:10.1371/journal.pone.0008250.
179. Tang, Y.; Zhang, Y.-Q.; Huang, Z. Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2007**, *4*, 365–381, doi:10.1109/TCBB.2007.70224.

180. Chen, X.-W.; Jeong, J.C. Enhanced recursive feature elimination. In Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA 2007), Cincinnati, OH, USA, 13–15 December 2007; pp. 429–435, doi:10.1109/ICMLA.2007.35.
181. Escanilla, N.S.; Hellerstein, L.; Kleiman, R.; Kuang, Z.; Shull J.; Page, D. Recursive Feature Elimination by Sensitivity Testing. In Proceedings of the 17th IEEE International Conference on Machine Learning and
Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 40–47, doi:10.1109/ICMLA.2018.00014.
182. Yeh, I.-C.; Cheng, W.-L. First and second order sensitivity analysis of MLP. *Neurocomp.* **2010**, *73*, 2225–2223, doi:10.1016/j.neucom.2010.01.011.
183. Garson, G.D. Interpreting neural-network connection weights. *AI Exp.* **1991**, *6*, 46–51.
184. Dimopoulos, Y.; Bourret, P.; Lek, S. Use of some sensitivity criteria for choosing networks with good generalization ability. *Neural Proc. Lett.* **1995**, *2*, 1–4, doi:10.1007/BF02309007.
185. Dimopoulos, I.; Chronopoulos, J.; Chronopoulou-Sereli, A.; Lek, S. Neural network models to study relationships between lead concentration in grasses and permanent urban descriptors in Athens city (Greece). *Ecol. Model.* **1999**, *120*, 157–165, doi:10.1016/S0304-3800(99)00099-X.
186. Lemaire, V.; Féraud, R. Driven forward features selection: A comparative study on neural networks. In Proceedings of the 13th International Conference, ICONIP 2006, Hong Kong, China, 3–6 October 2006; pp. 693–702, doi:10.1007/11893257_77.
187. Ding, S.; Li, H.; Su, C.; Yu, J.; Jin, F. Evolutionary artificial neural networks: A review. *Artif. Intell. Rev.* **2013**, *39*, 251–260, doi:10.1007/s10462-011-9270-6.
188. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. B* **1996**, *58*, 267–288.
189. Radchenko, P.; James, G.M. Improved variable selection with forward-LASSO adaptive shrinkage. *Ann. Appl. Stat.* **2011**, *5*, 427–448, doi:10.1214/10-AOAS375.
190. Tikhonov, A.N.; Goncharsky, A.; Stepanov, V.V.; Yagola, A.G. *Numerical Methods for the Solution of Ill-Posed Problems*; Springer: Berlin/Heidelberg, Germany, 1995.
191. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. B* **2005**, *67*, 301–320.
192. Sun, K.; Huang, S.-H.; Wong, D.S.-H.; Jang, S.-S. Design and Application of a Variable Selection Method for Multilayer perceptron Neural Network with LASSO. *IEEE Trans. Neual Netw. Learn. Syst.* **2017**, *28*, 1386–1396, doi:10.1109/TNNLS.2016.2542866.
193. Souza, F.A.A.; Araújo, R.; Matias, T.; Mendes, J. A multilayer-perceptron based method for variable selection in soft sensor design. *J. Proc. Control* **2013**, *23*, 1371–1378, doi:10.1016/j.jprocont.2013.09.014.
194. Fujiwara, K.; Kano, M. Efficient input variable selection for soft-sensor design based on nearest correlation spectral clustering and group Lasso. *ISA Trans.* **2015**, *58*, 367–379, doi:10.1016/j.isatra.2015.04.007.
195. Fujiwara, K.; Kano, M. Nearest Correlation-Based Input Variable Weighting for Soft-Sensor Design. *Front. Chem.* **2018**, *6*, doi:10.3389%2Ffchem.2018.00171.