From Sensor Readings to Predictions: on the Process of Developing Practical Soft Sensors

Marcin Budka[1], Mark Eastwood[2], Bogdan Gabrys[1],
Petr Kadlec[3], Manuel Martin Salvador[1], Stephanie Schwan[3],
Athanasios Tsakonas[1], and Indrė Žliobaitė[4]

[1] Bournemouth University, UK
{mbudka,bgabrys,msalvador,atsakonas}@bournemouth.ac.uk
[2] Coventry University, UK ab3276@coventry.ac.uk
[3] Evonik Industries, Germany {petr.kadlec,stephanie.schwan}@evonik.com
[4] Aalto University and HIIT, Finland indre.zliobaite@aalto.fi

Abstract. Automatic data acquisition systems provide large amounts of streaming data generated by physical sensors. This data forms an input to computational models (soft sensors) routinely used for monitoring and control of industrial processes, traffic patterns, environment and natural hazards, and many more. The majority of these models assume that the data comes in a cleaned and pre-processed form, ready to be fed directly into a predictive model. In practice, to ensure appropriate data quality, most of the modelling efforts concentrate on preparing data from raw sensor readings to be used as model inputs. This study analyzes the process of data preparation for predictive models with streaming sensor data. We present the challenges of data preparation as a four-step process, identify the key challenges in each step, and provide recommendations for handling these issues. The discussion is focused on the approaches that are less commonly used, while, based on our experience, may contribute particularly well to solving practical soft sensor tasks. Our arguments are illustrated with a case study in the chemical production industry.

1 Introduction

Automatic data acquisition systems, which are common nowadays, generate large amounts of streaming data. This data, provided by various physical sensors is used for monitoring and control of industrial processes, traffic patterns, environment and natural hazards to name a few. Soft sensors are computational models that aggregate readings of physical sensors to be used for monitoring, assessing and predicting the performance of the system. They play an increasingly important role in management and control of production processes [3, 7]. The popularity of soft sensors is boosted by increasing availability of real sensors, data storage and processing capacities, as well as computational resources. Soft sensors operate online using streams of sensor readings, therefore they need to be robust to noise and adaptive to changes over time. They also should use a limited amount of memory and be able to produce predictions in at most linear time with respect to data arrival.

Building soft sensors for streaming data has received a lot of attention in the last decade (see e.g. [7, 8]), often focusing on algorithmic aspects of the computational models, while the process of data preparation receives less attention in research literature [15]. Evidently, building a soft sensor is not limited to selecting the right model. In practice data preparation takes a lot of effort and often is more challenging than designing the predictive model itself. This paper discusses the process of building soft sensors with a focus on data preparation along with the case study from chemical industry. Our goal is to discuss the major issues of data preparation and experimentally evaluate the contribution of various data preparation steps towards the final soft sensor performance.

The main contribution of our study is a framework - a systematic characterization of data preparation process for developing industrial predictive models (soft sensors). Data preparation issue has received little attention in the research literature, while in industrial applications data preparation takes majority of the modelling time. In line with the framework we present our recommendations for data preparation that are based on our experience in building soft sensors within the chemical industry, and are illustrated with real data examples.

The paper is organised as follows. In Section 2 we discuss the requirements and expectations for soft sensors in chemical industry. Section 3 presents a framework for developing data driven soft sensors. In Section 4 we experimentally illustrate the role of three selected data preparation techniques in building accurate predictive models via a case study in the chemical production domain. Section 5 concludes the study, and discusses directions for future research.

2 Requirements and expectations for predictive models in the process industry

In the process industry soft sensors are used in four main applications: (1) *online prediction* of a difficult-to-measure variable from easy-to-measure variables; (2) *inferential control* in the process control loop; (3) *multivariate process monitoring* for determining the process state from observed measurements; and (4) as a *hardware sensor backup* (e.g. during maintenance).

This study focuses on the data-driven soft sensors for online predictions of difficult-to-measure variables. Many critical process values (e.g. the fermentation progress in a biochemical process, or the progress of polymerisation in a batch reactor) are difficult, if not impossible to measure in an automated way at a required sampling rate. Sometimes the *first-principle* models, that are based on the physical and chemical process knowledge, are available. Although such models are preferred by practitioners, they are primarily meant for planning and design of the processing plants, and therefore usually focus on the ideal states of the process. Thus, such models can seldom be used in practice in a wide range of operating conditions. Moreover, often the process knowledge for modelling is not available at all. In such cases data-driven models fill the gap and often play an important role for the operation of the processes as they can extract the process knowledge automatically from the provided data.

A successful soft sensor is a model, which has been implemented into the process online environment and accepted by the process operators. In order to gain acceptance the soft sensor has to provide reasonable performance, be stable and predictable. This means that performance has to be immune to changes often happening in the production plants. It is not uncommon that physical sensors fail, drift or become unavailable. Hence, the soft sensor needs to have some kind of automated performance monitoring and adaptation capability. The predictive performance is not the only success criterion however.

Transparency is another important property for model success. It is essential for the process operators to understand, how the soft sensor came to its conclusions. This becomes even more critical if the predictions deviate from the true value. In such cases, it is of utmost importance to be able to backtrack the wrong prediction to its cause. For this reason pure black-box methods like certain types of Artificial Neural Networks may have problems with gaining acceptance.

Another challenge is that after the model is deployed, it is often used by personnel with limited background in machine learning. Therefore, the operation of the model has to be completely automated, and as simple as possible in order to avoid frustration and resistance from the operating personnel.

A systematic approach for soft sensor development has been proposed in [9]. The authors present it as a four step process consisting of handling missing data, detecting and handling outliers, deriving a regression model and validating it on independent data. An alternative methodology, presented in [16], focuses on three steps: data collection and conditioning, selection of influential features and correlation building. In [12], in addition to a general three-step methodology a more specialised one, based on multivariate smoothing procedure, is also discussed. Its distinguishing feature is the focus on the collection of process knowledge, which is not evident in other approaches. Other general methodologies for soft sensor development have been proposed in [3, 6, 7] and [4], with the latter based on the Six-Sigma process management strategy.

The methodology presented in this paper builds upon some ideas proposed in the literature, augmented by our own experience and knowledge gained during interaction with process experts, plant operators and soft sensor practitioners.

3 A framework for developing data driven soft sensors

The framework describes soft sensor development process in four steps:

1. setting up the performance goals and evaluation criteria;
2. data analysis (exploratory);
3. data preparation and preprocessing:
   (a) data acquisition (deciding which data to collect and from which sources);
   (b) data preprocessing (de-noising, handling outliers and missing values);
   (c) data reduction (extracting relevant representation);
4. training and validating the predictive model.

We focus on the first three steps that have been understudied in data analysis literature. For model training and validation an interested reader is referred to one of the classical data mining or machine learning textbooks (e.g. [18]).

3.1 Setting up performance goals and evaluation criteria

When starting a soft sensor project we first need to define what the soft sensor is needed for and what will be the quantitative and qualitative evaluation criteria.

Qualitative evaluation. Many models are so called black-boxes, where it is difficult or impossible to track back the relation between the inputs and predictions. Knowing the effects of input features to the target is particularly important for controlling the process. Moreover, transparent models are typically more trusted and better accepted by the operators. Classification trees, regression trees, and nearest neighbour approaches are among the most transparent.

Computational requirements of the model need to be taken into account particularly in high throughput or autonomous systems operating on batteries.

Quantitative evaluation. The choice of appropriate error measure is critical. Not only it is important to choose a criterion, that is possible to optimize [1]. Even more important is that the criterion measures the performance aspects, that are practically relevant. The Root Mean Squared Error (RMSE) is very popular in research due to convenient analytical properties: $RMSE = \sqrt{1/n \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$, where y is the true target, $\hat{y}$ is the prediction and n is the size of the testing data. It punishes large deviations from the target, which is often very relevant for industrial applications, however, the meaning of this error may not be straightforward to interpret for the process experts and operators. The Mean Absolute Error (MAE) is often considered a more natural measure of average error [17]: $MAE = 1/n \sum_{i=1}^{n} |\hat{y}_i - y_i|$, but is more difficult to optimize.

Often, particularly in the control applications, predicting the direction of a signal change may be more important than low absolute deviation from the true value. In such cases it is useful to optimize the classification accuracy (CE). The accuracy is measured as a fraction of times the true signal from time $t-1$ to t goes to the same direction (up or down) as the prediction.

Variability of the predictions is critical in process control applications. A flat prediction is preferred to spiky, since following the latter would require very frequent process adjustments. Jitter (J) measures an average distance that one would travel if the prediction signal was followed: $J = \frac{1}{n-1} \sum_{i=2}^{n} |\hat{y}_t - \hat{y}_{t-1})|$, where n is the number of observations, observations need to be ordered in time.

Robustness and prediction confidence is understood as resistance of the predictor to impurities in the data, such as noisy, outlying or missing observations. If the system was to provide a completely wrong prediction, it should rather not produce any prediction at all. While some predictive methods, e.g. Gaussian Processes, inherently provide a confidence value, most of the them do not. A common approach is to generate an ensemble of models and use the disagreement of the individual models as a confidence estimate. Such an approach however, would carry higher computational costs and lower transparency of the predic-

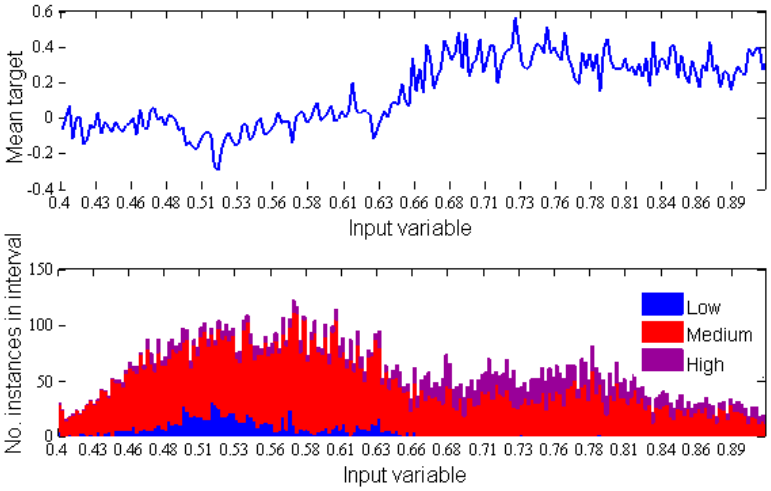

Fig. 1. An example of a profile plot — "low", "medium" and "high" are the target intervals.

tions. Alternatively, one can define a domain, where the model is applicable, and relate confidence values to the locations of test observations in this domain [11].

It is critical to understand the process and the potential role of the soft sensor as much as possible before deciding on which evaluation criteria to use.

3.2 Data analysis

When the objectives of a soft sensor have been decided upon, the next step is data understanding. The goal is to discover characteristic properties of the data that would help to build a more effective predictive model.

Exploratory data analysis can help to discover anomalies in data, define necessary data preparation approaches and determine potentially useful model classes. Scatterplots of variable against variable or against target are commonly used in exploratory analysis. In addition, we recommend to plot variables over time and to construct variable profiles by defining a small number of intervals on the target y and plotting a visualisation of where datapoints in each interval lie along a given input variable x.

To construct a profile we divide an input variable into a number of bins and find how many datapoints in each bin fall in each interval of the target. These are plotted in a stacked bar chart as in the lower part of Figure 1. This makes a crude representation of the density functions $P(y|x)$, and $P(x)$. Plotting the mean of y in each bin as in the upper part of Figure 1 provides further visualisation – in this example a profile in which the relationship is approximately monotonic.

Time series analysis involves the identification of stationarity in time series data, which assists in estimating the predictability of the time series, and

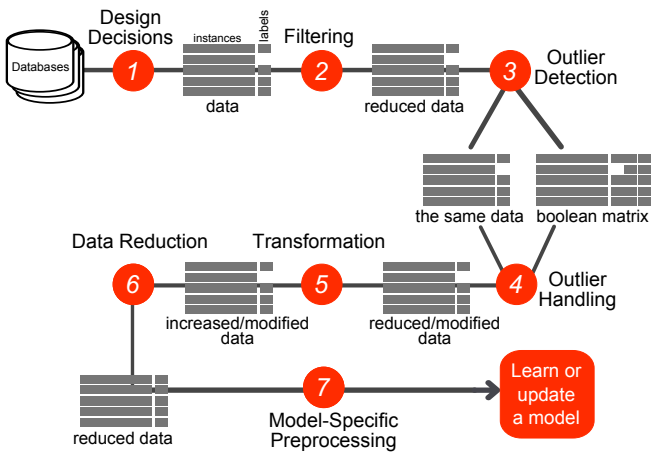

Fig. 2. The proposed order of data preprocessing steps.

can imply preferred forecasting methods. Any data-generating process is either stationary or integrated in an order higher than zero. Some empirically observed time series however exhibit dependencies between distant observations, and they are referred as *fractionally integrated processes* or *long-memory processes*. A process is considered to have long memory when the spectral density becomes unbounded at low frequencies. The most common approach to detect the presence of long-memory processes is the rescaled range statistic, commonly known as *R/S statistic*, which is directly derived by the *Hurst coefficient* [10].

We recommend to include domain experts in the data analysis from the beginning of the process in order to quickly detect potential problems of the available data.

3.3 Data preparation and preprocessing

Preprocessing transforms raw data into a format that can be more effectively used in training and prediction. Examples of preprocessing techniques include: outlier detection and removal, missing values replacement, data normalisation, data rotation, or feature selection. We leave out description of techniques, which can be found in machine learning textbooks, for instance [13, 18].

In industrial processes typically real time data processing is required. For autonomous operation preprocessing needs to be performed online, systematically, and design decisions need to be verifiable, therefore, the procedure and order of preprocessing actions need to be well defined. We propose a seven-step data preprocessing process, as highlighted in Figure 2.

The first step defines the data *design decisions*, such as, how data is queried from databases, how data sources are synchronized, at what sampling rate data arrives, whether any filtering (de-noising) is used. This step produces raw data in

a matrix form, where each row corresponds to an observation of sensor readings at one time. Typically, the design choices remain fixed during the project.

In industrial processes data often comes from multiple sources, which may be separated physically (e.g. a long flow pipe) or virtually (e.g. data is stored in different ways), and need to be synchronized. Synchronization of virtual sources requires consolidating the data into a single database or stream, and is relatively straightforward. Synchronizing data from different physical locations is usually more challenging, and can be estimated based on the physical properties of the process (e.g. speed of flow), or approached as as a computational feature selection problem, where for each sensor different time lags are tried as candidate features.

The second step *filters* out irrelevant data. For example, we can discard data during plant shut-down times and non-steady operation states. Rules for detecting such periods can be defined by experts during the design step, or statistical change detection techniques could be used to identify them automatically.

The third step *detects outliers* in four stages. Firstly, recordings are checked against physical constraints (e.g. water temperature cannot be negative). Secondly, univariate statistical tests can detect outliers in individual sensors. Thirdly, multivariate statistical tests on all variables together (see e.g. [2]) can detect outliers at an observation level. Finally, we check consistency of the instances with the target values. At this step outliers are flagged as missing values.

In the fourth step we *handle the identified outliers*, and/or missing values. In industrial applications predictions are needed continuously, therefore, removing observations with missing values is typically not an option. Standard missing value imputation techniques can be used, ranging from computationally light last observed value, or mean imputation, to various model based imputation methods (see e.g. [5]). The result is a modified data matrix, data size is still the same as produced in step 1.

The fifth step performs data *transformations*, which can modify the existing features (e.g. discretisation), derive new features (e.g. an indicator if the production is running), scale the data or rotate the data for de-correlation purposes. The result of this step is a data matrix that can have more or the same number of features than before and the same number of observations.

The sixth step *reduces data* by feature selection (or extraction) and observation selection (subsampling). As a result the data matrix will decrease in size.

The seventh step performs *model specific* preprocessing, e.g. further removing of outliers or undesired observations. This completes data preprocessing and we can proceed to model training.

While the design decisions (step 1) must be made, other steps (2-7) are optional, and it is up to data scientist to decide, which particular techniques to use. However, we recommend to keep the order as suggested. It enables reproducibility, allows easier documentation, and, most importantly, easier automation when it comes to implementation.

4 Case study

We have discussed the process of building soft sensors with an emphasis on data preprocessing, which in practice is often the essential step for obtaining good predictions. In this section we illustrate our arguments via a case study from chemical production domain. We experimentally analyze how selected data preprocessing steps contribute towards the accuracy of the final predictor.

We test the effects of data preparation on the real industrial dataset from a debutanizer column. The dataset covers three years of operation and consists of 189 193 records of 85 real-valued sensor readings. The sensors measure temperatures, pressures, flows and concentrations at various points in the process at every 5 min. The target variable is a real-valued concentration of the product.

We prepare 16 versions of data, which differ in preprocessing applied to them, while the final predictive model is the same. The idea is to compare the performance on these datasets pairwise, while within a pair all the preprocessing is the same, but one step is different. That allows to asses the contribution of that particular preprocessing step in isolation, other factors held constant.

Table 1. Versions of data with different preprocessing.

	Training size	Testing size	Subsampling	Synchronization	Feature selection	Fractal features	Difference data
RAW	188 752	21 859					
RAW-SYN	188 752	21 859		✓			
RAW-FET	188 752	21 859			✓		
RAW-SYN-FET	188 752	21 859		✓	✓		
SUB	15 611	1 822	✓				
SUB-SYN	15 611	1 822	✓	✓			
SUB-FET	15 611	1 822	✓		✓		
SUB-SYN-FET	15 611	1 822	✓	✓	✓		
SUB-DIF-FRA	15 610	1 822	✓			✓	✓
SUB-DIF-SYN-FRA	15 610	1 822	✓	✓		✓	✓
SUB-DIF	15 610	1 822	✓				✓
SUB-DIF-SYN	15 610	1 822	✓	✓			✓

The datasets are summarized in Table 1. **RAW** and **SUB** refer to different sampling rates. RAW uses data sampled at every 5 min, while SUB at every 1 h. **SYN** means that the input features of the data are synchronised by moving the features along the time axes to better reflect the physical process, as described in Section 3.3. **FET** indicates feature selection. We select 20 features that have the highest absolute correlation with the target variable from the original 85 features. We explore two options: early and late selection. Early (E) selection means that we select features from the first 1000 training examples. Late (L) selection means that we select features from the latest 1000 data points in the training set. If data is changing over time, we expect L to be more accurate. **FRA** means that

the space of the dataset has been complemented with the features derived by computing the fractal dimension, as presented in Section 3.2. Fractal features describe the Hurst exponent for each input and the output, calculated over the last 128 measurements. Finally, **DIF** refers to transformation of the input features and the target variable. Differenced data replaces the original values with the first derivative with respect to time. It describes how much the values are changing in comparison to the previous time step. For example, suppose y_t is the original variable at time t, then the differenced data is $r_t = y_t - y_{t-1}$.

The experimental protocol is as follows. Size of the training and testing sets for each dataset are reported in Table 1. For testing we use a hold out set, which did not participate in the parameter tuning of the preprocessing methods. The performance is evaluated using the mean absolute error (MAE). The predictions on DIF datasets are transformed back to the original space before measuring the error. We use the Partial Least Squares regression (PLS) [14] as the predictive model with the number of hidden variables set to 10.

Tables 2 and 3 present the results in MAE, the lower - the better. Preprocessing actions are assessed pairwise - the baselines are on the left, and datasets with additional preprocessing are on the right. Table 2 covers non-differentiated data, and Table 3 presents the results on differentiated data. We treat these cases separately, since differentiating allows to capture different properties of the signal (autoregressive properties), and hence leads to different errors.

Table 2. Testing errors (MAE) on non-differentiated data (• - superior performance).

preprocessing#1	MAE#1	MAE#2	preprocessing#2	improvement
RAW	225	222 •	RAW-SYN	3 (1%)
SUB	227	221 •	SUB-SYN	6 (3%)
RAW-FET-E	228	198 •	RAW-FET-L	30 (13%)
RAW-SYN-FET-E	245	201 •	RAW-SYN-FET-L	44 (18%)
SUB-FET-E	236	193 •	SUB-FET-L	43 (18%)
SUB-SYN-FET-E	215	185 •	SUB-SYN-FET-L	30 (14%)

We see that each selected preprocessing action improves the predictive performance (the right approaches are better than the left). The largest improvement is achieved by late feature selection (RAW-SYN-FET-L and SUB-FET-L), as compared to early feature selection. This is an interesting observation. This suggests, that feature relevance is changing over time, and we can achieve as much as 18% reduction in the prediction error only by making the feature selection adaptive over time using the simplest fixed sliding window strategy.

Table 3 presents the mean absolute errors (MAE) of the predictions.

Comparing the results on non-differentiated data in Table 2 and differentiated data in Table 3 suggests that taking into account self-similarity is very beneficial. That is not surprising, considering that in chemical production processes operating conditions do not jump suddenly, hence, the concentration of

Table 3. Testing errors (MAE) on differentiated data (• - superior performance).

preprocessing#1	MAE#1	MAE#2	preprocessing#2	improvement
SUB-DIF	41.8	35.3 •	SUB-DIF-SYN	6.5 (16%)
SUB-DIF	41.8	32.4 •	SUB-DIF-FRA	9.4 (22%)

the output also remains similar to what has been observed in the recent past, therefore, methods from time series modeling contribute well.

Experiments show that preprocessing actions consistently improve the predictive performance, with adaptive feature selection making the largest impact.

In Section 3.1 we also discussed qualitative criteria, namely transparency and computational load. In terms of computational load, PLS regression that was used can be updated recursively using analytical solutions, it does not require optimization loops, and is easy to handle on a commodity hardware.

PLS regression is one of the most transparent models available. While fitting the model is somewhat more involved, the result is a linear model. The coefficients at the inputs can be interpreted as the importance weights, hence, PLS regression provides good transparency and interpretability, easy to understand even for non-experts. All the tested preprocessing actions, except maybe the fractal dimension, do not reduce transparency and interpretability in any substantial way. For example, synchronization of the features (SYN) shifts observations in time, but the regression coefficients remain as interpretable, as before.

5 Conclusion

We analyzed data preparation process for building soft sensors in three main steps: establishing the evaluation framework, exploratory data analysis and data preparation. We recommended a sequence of data preparation techniques for building soft sensors. We illustrated our propositions with a case study with real data from industrial production process. The experiments showed that the selected preprocessing actions consistently improve the predictive performance, and adaptive feature selection makes the largest contribution towards improving the prediction accuracy.

This study opens several interesting directions for further research. Firstly, since relational and autoregressive data representations capture different patterns in data, combining the two approaches suggests a promising research direction. A straightforward way to combine would be to extend the feature space with autoregressive features. Alternatively, we could combine those different types of approaches into an ensemble for the final decision making.

Secondly, it would be interesting to explore how to filter out the effects of data compression when evaluating predictive models. One direction is to identify the real sensor readings and treat the compressed readings as missing values. It would be also interesting to analyse theoretically what compression algorithms would be the most suitable for streaming data, which is later intended to be used for predictive modelling.

Finally, despite thorough synchronisation of preprocessing steps we encountered different data representations (after each preprocessing step), not straightforward to integrate. Thus, our study confirmed the intuition that automating and combining multiple data preparation methods into a single autonomous system that would use a feedback loop to update itself is urgently needed.

Acknowledgment

Thanks Evonik Industries for providing the data. The research leading to these results has received funding from the European Commission within the Marie Curie Industry and Academia Partnerships & Pathways (IAPP) programme under grant agreement no. 251617.

References

1. M. Budka. Clustering as an example of optimizing arbitrarily chosen objective functions. In *Advanced Methods for Comp. Collective Intell.*, pages 177–186. 2013.
2. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, July 2009.
3. L. Fortuna. *Soft sensors for monitoring and control of industrial processes*. Springer, 2007.
4. C. Han and Y. Lee. Intelligent integrated plant operation system for six sigma. *Annual Reviews in Control*, 26(1):27–43, 2002.
5. R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data, 2nd Edition*. Wiley, 2002.
6. P. Kadlec and B. Gabrys. Architecture for development of adaptive on-line prediction models. *Memetic Computing*, 1(4):241–269, 2009.
7. P. Kadlec, B. Gabrys, and S. Strandt. Data-driven soft sensors in the process industry. *Computers and Chemical Engineering*, 33(4):795–814, 2009.
8. P. Kadlec, R. Grbic, and B. Gabrys. Review of adaptation mechanisms for data-driven soft sensors. *Computers & Chemical Engineering*, 35(1):1–24, 2011.
9. B. Lin, B. Recke, J. Knudsen, and S. Jorgensen. A systematic approach for soft sensor development. *Computers & chemical engineering*, 31(5-6):419–425, 2007.
10. B. Mandelbrot. *The fractal geometry of nature*. W.H. Freeman, 1983.
11. T. Netzeva, A. Worth, T. Aldenberg, R. Benigni, M. Cronin, P. Gramatica, J. Jaworska, S. Kahn, G. Klopman, C. Marchant, et al. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. *Alternatives to Laboratory Animals*, 33(2):1–19, 2005.
12. S. Park and C. Han. A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns. *Computers & Chemical Engineering*, 24(2-7):871–877, 2000.
13. R.K. Pearson. Mining imperfect data. *Society for Industrial and Applied Mechanics, USA*, 2005.
14. J. Qin. Recursive PLS algorithms for adaptive data modeling. *Computers & Chemical Engineering*, 22(4-5):503–514, 1998.
15. I. Žliobaitė and B. Gabrys. Adaptive preprocessing for streaming data. *IEEE Trans. on Knowledge and Data Engineering*, 26:309–321, 2014.

16. K. Warne, G. Prasad, S. Rezvani, and L. Maguire. Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion. *Eng. Appl. of Artif. Intell.*, 17(8):871–885, 2004.
17. C. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30:79–82, 2005.
18. I. Witten, E. Frank, and M. Hall. *Data Mining: Practical Machine Learning Tools and Techniques (3rd Ed.)*. Morgan Kaufmann, 2011.