Preprints of the
9th International Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
June 7-10, 2015, Whistler, British Columbia, Canada

Fault Identification in Batch Processes Using Process Data or Contribution Plots: A Comparative Study ⋆

Sam Wuyts * **Geert Gins** * **Pieter Van den Kerkhof** *
Jan Van Impe *

* *KU Leuven department of Chemical Engineering, Chemical & Biochemical Process Technology & Control (BioTeC), Willem de Croylaan 46, 3001 Leuven (e-mail: {geert.gins, sam.wuyts, jan.vanimpe}@cit.kuleuven.be).*

Abstract: In statistical process monitoring, contribution plots are commonly used by operators and experts to identify the root cause of abnormal events. Because contribution plots suffer from fault smearing – an effect that possibly masks the cause of an upset – this paper investigates whether automated fault identification can be improved by using process data instead of contributions. Hereto, both approaches (i.e., using either the sensor measurements or their contributions as inputs for a classification model) are tested on the benchmark penicillin fermentation process `Pensim`, implemented in `RAYMOND`. To optimize the performance of each approach, different manipulations of both the process data and the variable contributions are introduced based on the nature of the occurring faults. It is observed that these manipulations have a large influence on the classification performance. Furthermore, this paper demonstrates that fault smearing negatively affects the classification based on the variable contributions. It is concluded that automated fault identification is improved by using the process data rather than the variable contributions as model inputs for the case study investigated.

Keywords: Batch processes; Fault identification; Statistical Process Control (SPC); Classification models; Contribution plots

1. INTRODUCTION

Compared to continuous processes, batch processes exhibit low investment costs and are able to reach higher conversions. Furthermore, batch processes have a larger flexibility as one reactor can produce different (grades of) products. Therefore, batch processes play an important role in the (bio)chemical industry, especially in the production of high added value products such as pharmaceuticals or specialty chemicals. Batch processes are inherently transient in nature. Furthermore, online measurements or estimates of the *final* quality – which is of interest rather than instantaneous quality, if the latter can even be defined – are seldom available. These present important challenges in batch monitoring, control, fault detection and diagnosis.

Statistical Process Control (SPC) offers a solution for better monitoring and control of batch processes. Its development is supported by the large historical databases that chemical plants typically possess, containing frequent measurements of online sensors on a large set of process variables. SPC exploits these databases to enable better monitoring and control and, hence, faster fault detection.

⋆ Work supported in part by Project PFV/10/002 (OPTEC Optimization in Engineering Center) of the Research Council of the KU Leuven, Project KP/09/005 (SCORES4CHEM) of the Industrial Research Council of the KU Leuven, and the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian Federal Science Policy Office. The authors assume scientific responsibility.

After the detection of a fault, the task remains to isolate and identify the source of the disturbance. Compared to fault detection, fault isolation is a much more complex task which is not always feasible. For fault identification, contribution plots are typically used to facilitate the diagnosis by narrowing the search region of possible root causes (Westerhuis et al., 2000). Contribution plots require no prior knowledge about underlying disturbances but do not always unequivocally point out the variable(s) causing the fault. Therefore, expert knowledge is still required for correct interpretation of the contribution pattern.

When all possible fault classes are known, fault identification reduces to fault classification. An automated classifier is trained on faulty data and searches the most probable root cause of the detected upset. The required interpretation of the contribution pattern is bypassed and therefore the time between fault detection and corrective action is significantly reduced. However, the so-called *fault smearing effect* still exerts a negative influence on the accuracy of the contributions, possibly resulting in misclassification (Van den Kerkhof et al., 2013; Westerhuis et al., 2000).

For fast and reliable automatic classification, it might be rewarding to look for possible alternatives to the contribution plots, which are subject to the fault smearing effect. This paper focuses on the comparison of the classification performance following two different approaches: the first provides the process data (i.e., raw or pretreated, but

without any projection) to the classification models while the second entails the classification based on the variable contributions obtained from a PCA model. To optimize performance for both approaches, different manipulations of both process data and variable contributions are introduced based on the nature of the occurring faults.

The remainder of this paper is organized as follows. Section 2 summarizes the typical monitoring procedure for fault detection and identification. Two classification models, k Nearest Neighbors (k-NN) and Least Squares Support Vector Machines (LS-SVM), are briefly introduced in Section 3. Section 4 describes the case study on which the fault detection and classification routine is tested, followed by a discussion of the results obtained in Section 5.

2. FAULT DETECTION & IDENTIFICATION

A three-step procedure is used to detect and identify abnormal operation (MacGregor and Kourti, 1995). The first step in the approach consists of constructing a mathematical model that characterizes normal operation (Section 2.1). Next, current observed behavior of the installation is checked against this reference model to detect abnormal behavior via fault detection statistics (Section 2.2). Finally, contribution plots are analysed to identify the root cause of each identified disturbance (Section 2.3).

2.1 Principal Components Analysis (PCA)

Principal Component Analysis (PCA; Jolliffe, 1986) is used to model Normal Operation Conditions (NOC) of the process by characterizing the correlation between the different sensors during periods of good operation. Effectively, PCA identifies a small number of *latent* variables that characterize the basic operation of the installation.

Data from I NOC batches is collected as training data. In each batch, J sensors are sampled on K time points. Each sensor is scaled to zero mean and unit variance around its average trajectory over the I batches. Next, data are arranged using variable-wise unfolding (Wold et al., 1998): the $K \times J$ data matrices for all batches are stacked in an $(IK \times J)$ data matrix $\mathbf{X}$. PCA summarizes $\mathbf{X}$ in a smaller number of R uncorrelated scores $\mathbf{T}$ $(IK \times R)$. Hereto, it finds the directions of maximal variation in $\mathbf{X}$ as the eigenvectors of the covariance matrix $\mathbf{X}^\top\mathbf{X}$. The R leading eigenvectors explain most of the variation in $\mathbf{X}$ and are stored in the loading matrix $\mathbf{P}$ $(J \times R)$. The matrix $\mathbf{E}$ $(IK \times J)$ contains the residuals.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} \tag{1}$$

$$\mathbf{T} = \mathbf{X}\mathbf{P} \tag{2}$$

The fraction of the total variation explained by component r is proportional to its corresponding eigenvalue λ_r.

$$f_r = \frac{\lambda_r}{\sum_{i=1}^{J}\lambda_i} \tag{3}$$

The number of principal components R is determined by the user. Many criteria exist. In this work, an adjusted Wold criterion (Li et al., 2002) is used on the cumulative fraction of explained variance.

$$R = \min\left(R \;\middle|\; \frac{\sum_{r=1}^{R+1} f_r}{\sum_{r=1}^{R} f_r} \leq 1.05\right) \tag{4}$$

2.2 Fault Detection Statistics

When operating a new batch online, two statistical measures compare how closely a new set of measurements $\mathbf{x}_k$ $(1 \times J)$ at time k matches NOC conditions.

Hotelling's T^2 statistic computes the distance in the model plane from the new scores at time k, $\mathbf{t}_k = \mathbf{x}_k\mathbf{P}$ $(1 \times R)$, to the region spanned by NOC scores $\mathbf{T}_k$ $(I \times J)$ at the same time k to detect extrapolation. The Q statistic monitors the residuals of the PCA model to detect model mismatch.

$$T^2(k) = \mathbf{t}_k\left(\frac{\mathbf{T}_k^\top\mathbf{T}_k}{I-1}\right)^{-1}\mathbf{t}_k^\top \tag{5}$$

$$Q(k) = \left(\mathbf{x}_k - \mathbf{t}_k\mathbf{P}^\top\right)\left(\mathbf{x}_k - \mathbf{t}_k\mathbf{P}^\top\right)^\top \tag{6}$$

An upper control limit with significance level α is determined from the training data for each statistic.

$$UCL_{T^2} = \frac{R\left(I^2-1\right)}{I\left(I-R\right)}F\left(R, I-R; \alpha\right) \tag{7}$$

$$UCL_Q = \frac{\sigma_k^2}{2m_k}\chi^2\left(2m_k^2/\sigma_k^2; \alpha\right) \tag{8}$$

m_k and σ_k^2 are the mean and variance of Q at time k. $F(R, I-R; \alpha)$ and $\chi^2(2m_k^2/\sigma_k^2; \alpha)$ are critical values at tolerance level α of, respectively, an F-distribution with R numerator and $I-R$ denominator degrees of freedom and a chi squared distribution with $2m_k^2/\sigma_k^2$ degrees of freedom. To improve robustness, crossvalidation is used to estimate $\mathbf{T}_k$, m_k, and σ_k^2.

2.3 Fault Identification via Contributions

After detecting an upset, contribution plots are used to identify the root cause. In this paper, two commonly-used contribution types are employed to compute variable j's contribution to the T^2 or Q statistic at time k: Complete Decomposition Contributions (CDC) and Reconstruction Based Contributions (RBC) (Alcala and Qin, 2011).

$$CDC_{jk} = \left(\boldsymbol{\xi}_j^\top\mathbf{M}^{\frac{1}{2}}\mathbf{x}_k^\top\right)^2 \tag{9}$$

$$RBC_{jk} = \frac{\left(\boldsymbol{\xi}_j^\top\mathbf{M}\mathbf{x}_k^\top\right)^2}{\boldsymbol{\xi}_j^\top\mathbf{M}\boldsymbol{\xi}_j} \tag{10}$$

$\boldsymbol{\xi}_j$ is the j-th column of the $J \times J$ identity matrix. $\mathbf{M}$ is statistic-dependent.

$$\mathbf{M}_{T^2} = \mathbf{P}\left(\frac{\mathbf{T}_k^\top\mathbf{T}_k}{I-1}\right)^{-1}\mathbf{P}^\top \tag{11}$$

$$\mathbf{M}_Q = \mathbf{I} - \mathbf{P}\mathbf{P}^\top \tag{12}$$

When monitoring a new batch, new data are projected on the model space, compressing measurements in a small number of scores. During calculation of the contributions, information is again extracted from the limited number of score values. This compression and expansion leads to each faulty measurement influencing all scores and the reconstruction of all other variables (Westerhuis et al., 2000). This fault smearing between contributions possibly results in wrong diagnosis (Alcala and Qin, 2011; Van den Kerkhof et al., 2013).

3. CLASSIFICATION MODELS

Automated classification models eliminate the subjective and time consuming interpretation of faults. The selection of the best performing classification model falls beyond the scope of this paper. Therefore, only two different model types are employed: k-NN and LS-SVM. The former excels in simplicity while the latter is a very powerful yet more complex nonlinear classifier with excellent generalization properties (Suykens et al., 2002; Abe, 2005).

3.1 k Nearest Neighbors

The k-NN method (Fix and Hodges, 1951) is one of the simplest classification models. Based on a certain measure of distance (often the Euclidean distance) between a new pattern and each pattern in the training set, the k closest neighbors are determined. The training set typically consists of multidimensional vectors, each classified with a label. A new pattern is assigned to the class that is most strongly represented among the k nearest neighbors. If the votes in the k closest neighbors tie between multiple classes, the pattern is considered to be unclassifiable. Despite its simplicity, k-NN has proven to be a powerful classification tool (King et al., 1995; Wu et al., 2008). In this work, $k = 1$ and $k = 3$ are tested.

3.2 Least Squares Support Vector Machines

The basics of LS-SVM and their extension towards multi-class classification are briefly discussed here. A more detailed elaboration concerning Support Vector Machines is presented in the book of Suykens et al. (2002).

LS-SVM basics An LS-SVM is a binary classifier for an M-dimensional input pattern vector $\mathbf{z}$ $(M \times 1)$ based on a training set of N training patterns $\mathbf{z}_n$ $(n = 1 \dots N)$. The training vectors are labeled with a scalar $y_n \in \{-1, +1\}$ for the positive and negative class respectively.

In the linear case, LS-SVMs assign a class to a new input vector $\mathbf{z}$ by formulating a decision function

$$y = \operatorname{sgn}\left(\mathbf{w}^T\mathbf{z} + b\right) \tag{13}$$

with coefficients $\mathbf{w}$ $(M \times 1)$ and bias b. The margin between the positive and negative classes is maximized to optimize generalization performance.

The decision function must be linear in its parameters, but can still be used for nonlinear classification (Boser et al., 1992). Hereto, the input pattern $\mathbf{z}$ is mapped to a higher-dimensional feature space using a nonlinear map $\boldsymbol{\varphi}(\cdot)$. Aizerman et al. (1964) implicitly define $\boldsymbol{\varphi}(\cdot)$ using kernel functions $K(\mathbf{z}_n, \mathbf{z}) = \boldsymbol{\varphi}(\mathbf{z}_n)^\top \boldsymbol{\varphi}(\mathbf{z})$. Mercer's theorem is used to rewrite Eq. 13 in its kernel form.

$$y = \operatorname{sgn}\left(\sum_{n=1}^{N} \alpha_n y_n K(\mathbf{z}_n, \mathbf{z}) + b\right) \tag{14}$$

The coefficients α_n and bias b are obtained by solving a set of linear equations with equality constraints (Ye and Xiong, 2007). Nonlinear classification is enabled by selecting nonlinear kernel functions. In this paper, a simple linear kernel and the commonly used Radial Basis Function (RBF) kernel are employed.

$$K(\mathbf{z}_n, \mathbf{z}) = \mathbf{z}_n^\top \mathbf{z} \qquad \text{linear kernel} \tag{15}$$

$$K(\mathbf{z}_n, \mathbf{z}) = \exp\left(-\frac{||\mathbf{z}_n - \mathbf{z}||_2^2}{\sigma^2}\right) \qquad \text{RBF kernel} \tag{16}$$

$||\mathbf{z}_n - \mathbf{z}||_2^2$ is the squared Euclidean distance between $\mathbf{z}_n$ and $\mathbf{z}$ while σ^2 is a parameter called the kernel width.

Multi-class LS-SVM Fault identification is a multi-class problem because it typically involves more than two types of faults. Therefore, the classification problem is decomposed into a series of binary classifications using binarization. Two popular binarization methods are *One-versus-One* (OvO) and *One-versus-All* (OvA) (Hastie and Tibshirani, 1998). Both are employed in this work.

OvO trains a binary classifier between each possible pair of classes. Only training patterns of the two classes involved are used for the establishment of the classifier, the other patterns are ignored. A new pattern is assigned to the class that wins the most pairwise comparisons. If multiple classes have an equal score, $\mathbf{z}$ is considered unclassifiable. Because each binary classifier assigns a new pattern to one of the two classes it was trained on, many binary classifiers erroneously classify the new pattern, possibly leading to incorrect final classification (Furnkranz, 2002).

OvA uses a binary classifier for each of the C classes, classifying the class members as positive and all other patterns as negative. If a new vector $\mathbf{z}$ is assigned to multiple or none of the classes, it is considered unclassifiable. OvA only requires C binary classifiers, which is less compared to OvO ($C(C-1)/2$ binary classifiers). Each binary OvA classifier uses all the training vectors available. However, OvA binarization results in more complex class boundaries and larger unclassifiable regions (Gins et al., 2015).

Practical Implementation The `LS-SVMlab` toolbox [1] for MATLAB is used to construct the multi-class LS-SVMs (Suykens et al., 2002). The optimal model parameters are determined via a two-step procedure that minimizes the classification error in crossvalidation. Coupled Simulated Annealing yields an initial parameter estimate that is subsequently fine tuned with a simplex method.

4. CASE STUDY

The methods presented in this paper are applied to data from an industrial-scale production of penicillin, generated by an extended version of the `Pensim` simulator developed by Birol et al. (2002) and implemented in `RAYMOND` [2] (Gins et al., 2014).

The process consists of two phases. During the initial batch phase, the micro-organisms are grown to a sufficiently high concentration. When the substrate concentration drops below 0.3 g/L, the fed-batch phase is initiated and new substrate is continuously fed at a constant rate of 0.06 L/h to stimulate production of penicillin. The fermentation is terminated when 25 L of substrate have been fed.

[1] Available online at `http://esat.kuleuven.be/sista/lssvmlab`
[2] Available online at `http://cit.kuleuven.be/biotec/raymond`

Table 1. Online measurements available as `Pensim` simulation output

Time	Feed rate
Dissolved oxygen (DO)	Agitator power
Fermentation volume	Feed temperature
Dissolved CO_2	Coolant flow rate
pH	Base flow rate
Reaction temperature	Acid flow rate

Table 2. Process faults with their relative magnitudes and starting times

Fault type	Magnitude	Onset [h]
Feed concentration step	$\pm[1\%, 10\%]$	0
Coolant temperature step	$\pm[1\%, 10\%]$	0
Agitator power drop	$-[5\%, 30\%]$	$20 - 380$
Aeration rate drop	$-[70\%, 90\%]$	$20 - 380$
Feed rate drift	$\pm[0.15\%/h, 0.35\%/\text{h}]$	$70 - 380$
DO sensor drift	$\pm[0.50\%/h, 0.75\%/\text{h}]$	$20 - 380$

A selection of 12 measurement variables, listed in Table 1, is retained from the simulation. Their time signals are aligned to 101 and 501 samples in, respectively, the batch and the fed-batch phase using indicator variables (Birol et al., 2002). Only basic measurement noise as in Birol et al. (2002) is considered to focus on the inherent diagnosability of the process upsets.

A total of 200 in-control batches (NOC) with varying initial conditions is simulated to obtain representative data of normal process behavior. Data pretreatment encompasses scaling around the average time trajectory to unit variance and zero mean, followed by variable-wise unfolding. A separate PCA fault detection model is constructed for each phase. The adjusted Wold criterion (Eq. 4) selects 6 principal components (92% of the total variance) for the batch phase and 8 principal components (93% of the total variance) for the fed-batch phase.

For each of the faults listed in Table 2, 1000 batches are generated for the training of the classification models yielding a total of 6000 faulty batches. The relative magnitude and time of occurrence of each fault is randomly picked from a uniform distribution with bounds in accordance with Table 2. For step and drift faults, an equal number of up- and downwards steps/drifts is simulated. Classifiers are trained using sets of 12–60 training batches (2–10 per fault class) randomly sampled from the available batches. A separate set of 600 validation batches (100 batches per fault type) is generated for validation of the classifiers. Due to space limitations, only results for a training set size of 6 batches per fault class are presented here.

In this paper, a closed set of six known fault types is considered (Table 2). A local novelty detection framework (e.g., Bodesheim et al., 2015) can be employed to detect new fault types or combinations of known faults. Next, an extra fault class can be defined and the classifier model updated.

The training and validation batches are monitored with the NOC PCA model. The process measurements and their variable contributions at time of detection are used for classification. To ensure good distinction between the different fault classes for classifier training, batches with false alarms are disregarded.

This procedure is performed 100 times to obtain statistically relevant results, each time using a different training set. For LS-SVM classification, the classification for each set of training batches is repeated an additional 100 times per training set to even out the performance spread caused by the probabilistic optimization routines during LS-SVM tuning (Suykens et al., 2002).

5. RESULTS

In order to optimize the performance of the classification, different pretreatments of the information (process data or contributions) provided to the classification models are considered. These pretreatments are chosen based on process insight and the nature of the faults occurring and aim to maximize the classification performance. The mean values and standard deviations of the class-specific correct 1-NN classification rates for the different pretreatments are depicted in Table 3. The results for the process data are discussed in Section 5.1; Section 5.2 elaborates on the results for the contributions. Global results for other classification models are covered in Section 5.3.

5.1 Classification using process data

The simplest classification option ("Raw") uses the raw new measurement $\mathbf{x}_k$ at the moment of detection as input for the classifier: $\mathbf{z} = \mathbf{x}_k^\top$. The poor classification rates are due to the transient nature of batch processes: because batch processes are never in steady state, the raw data pattern $\mathbf{x}_k$ depends on the moment of detection. Therefore, the spread in detection moments introduces an extra uninformative variability and hence complicates correct classification. For the feed concentration and coolant temperature steps, there is no spread of detection moments and therefore their classification is very accurate.

To eliminate the effect caused by the spread in detection moments, the next pretreatment ("Normalized") normalizes the measurements around their mean time profiles prior to classification, resulting in increased performance.

Working with absolute values of normalized measurements ("Absolute") eliminates the differences between positive and negative drifts or steps and increases the uniformity within one fault class. However, classification accuracy decreases for aeration rate drops and DO sensor drifts. These faults are both most pronounced in the DO measurement. Therefore, taking absolute values increases the similarity between the patterns of the aeration rate drops (always negative) and DO sensor drifts (positive or negative), resulting in misclassification between both classes.

For better distinction between aeration rate drops and DO sensor drifts, the time profiles of the measurement vectors are taken into account ("Window"). For a window L, this corresponds to supplying a $1 \times (L+1)J$ vector containing the measurements for time points $k \dots k+L$ to the classification model: $\mathbf{z} = [\mathbf{x}_k \ \mathbf{x}_{k+1} \ \dots \ \mathbf{x}_{k+L}]^\top$. In this work, a window of $L = 10$ is used. The time window facilitates the discrimination between faults that are drifts and faults that are sudden steps or drops and hence increase classification performance.

Table 3. Class-specific correct classification rates using 1-NN classification

Process data	Raw		Normalized		Absolute		Window		Outer	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Feed conc. step	96.7%	6.2%	100.0%	0.0%	100.0%	0.0%	100.0%	0.0%	100.0%	0.0%
Coolant temp. step	100.0%	0.0%	90.2%	5.1%	90.3%	5.3%	97.1%	2.6%	95.1%	4.1%
Agitator power drop	28.7%	6.3%	99.3%	2.3%	99.6%	1.6%	99.5%	1.7%	99.5%	1.7%
Feed rate drift	32.0%	9.2%	97.1%	8.6%	100.0%	0.2%	100.0%	0.3%	99.9%	0.6%
Aeration rate drop	24.0%	7.0%	72.0%	14.8%	55.9%	10.8%	98.8%	3.1%	99.4%	1.9%
DO sensor drift	26.2%	6.3%	69.7%	11.0%	63.3%	12.9%	94.9%	2.4%	94.6%	2.0%
Global	51.3%	2.0%	88.0%	2.7%	84.9%	2.1%	98.4%	0.6%	98.1%	0.8%

Q **statistic CDC**	Raw		Rescaled		Absolute		Window		Outer	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Feed conc. step	100.0%	0.0%	100.0%	0.0%	—	—	100.0%	0.0%	100.0%	0.0%
Coolant temp. step	88.1%	6.0%	86.8%	6.0%	—	—	87.6%	6.7%	91.8%	7.9%
Agitator power drop	74.8%	8.7%	77.5%	10.5%	—	—	89.4%	10.5%	93.9%	6.7%
Feed rate drift	76.3%	13.8%	76.3%	12.1%	—	—	78.5%	9.8%	86.7%	8.0%
Aeration rate drop	59.0%	12.0%	55.4%	11.6%	—	—	98.6%	1.9%	98.1%	3.7%
DO sensor drift	60.0%	13.5%	62.6%	13.9%	—	—	94.2%	2.1%	94.4%	1.7%
Global	76.4%	3.0%	76.4%	3.5%	—	—	91.4%	2.4%	94.2%	2.0%

A final improvement of classification accuracy might be to only provide the first and last time points of a time window to the classification model ("Outer"): $\mathbf{z} = [\mathbf{x}_k \ \mathbf{x}_{k+L}]^\top$. This reduces the redundancy in the classifier input and hence provides better conditioned data to the classifier. However, no significant improvement is observed.

5.2 Classification using contributions

Similar pretreatments as for the process data are applied to the contributions. Due to space limitations, only the results for the CDC towards the Q statistic are discussed, but the results for other contributions show the same general trends. The raw contributions ("Raw") already contain more relevant information compared to the raw process data. This is because the raw process data experience a large variability due to the spread in detection moments. In the contributions, this extra variability is not present since the normalization around the average time profile – and, hence, removal of most dynamic effects – is already carried out prior to projection of the data on the PCA model space to calculate the contributions.

Zhao et al. (2008) state that correct and reliable classification is only possible when the absolute contributions for faulty batches are compared with respect to their values under normal operation. Therefore, a rescaling of the contributions is considered ("Rescaled"), where the contributions are centered and scaled with the mean and standard deviation of the (cross-validated) NOC contributions. The classification performance does not significantly increase for this preprocessing method. This is expected since, as stated above, normalization around the average time profile is carried out prior to calculation of the contributions. Therefore, this pretreatment is not considered in the remainder of this paper.

Because CDC and RBC are positive by definition, taking their absolute values ("Absolute") is not considered.

To improve classification using contributions, time windows of variables' contributions rather than single time points can be employed (Ündey et al., 2003). Similar as for the process data, using time windows of *raw* contributions ("Window" and "Outer") indeed improves classification performance by making a better distinction between drifts and sudden steps or drops.

When comparing these results with those obtained in Section 5.1, it is clear that the highest achievable performance using the contribution plots is significantly below the performance that can be achieved using classification models that are trained on the process data.

5.3 Other classification models

For 3-NN and LS-SVM classification, only the global classification rates are presented in Table 4; their class-specific rates are not discussed due to space limitations. Numerical problems were encountered during the identification of LS-SVM models with RBF kernels using contributions as inputs. To ensure comparability and understandability of the trends in the classification rates, these cases are omitted. The same general trends as for 1-NN classification are observed: increasing performance from raw data ("Raw") to time windows of (pretreated) measurement vectors ("Window" and "Outer").

Other observed trends are the higher performance of k-NN compared to LS-SVM and of OvO compared to OvA binarization. It is hypothesized that the unbalanced nature of data set and/or the more complex class boundaries that are required to separate the OvA cases lead to this lower generalization performance.

6. CONCLUSION

This paper compares the fault identification performance of classification models using process data as model inputs to that of classifiers based on variable contributions. From the `Pensim` case study, it is concluded that fault identification is best performed using the process data rather than the variable contributions as model inputs.

However, it was observed that data pretreatment also affects classifier performance to a very large extent. It is,

Table 4. Global correct classification rates for 3-NN and LS-SVM classification models

Process data	Raw		Normalized		Absolute		Window		Outer	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
3-NN	43.7%	2.6%	84.2%	3.9%	84.0%	2.1%	97.4%	1.0%	96.5%	1.4%
LS-SVM linear kernel OvO	67.4%	4.7%	74.1%	4.9%	84.3%	2.7%	96.0%	2.1%	96.6%	1.9%
LS-SVM linear kernel OvA	57.6%	5.8%	67.7%	7.2%	73.9%	4.1%	90.3%	4.8%	96.4%	2.0%
LS-SVM RBF kernel OvO	51.4%	5.4%	85.2%	4.2%	81.9%	3.7%	94.1%	5.1%	94.2%	3.6%
LS-SVM RBF kernel OvA	39.7%	6.5%	76.5%	8.5%	73.3%	8.8%	86.0%	10.8%	91.0%	7.7%

Q **statistic CDC**	Raw		Rescaled		Absolute		Window		Outer	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
3-NN	74.0%	4.0%	75.7%	4.4%	—	—	85.7%	3.4%	88.8%	3.6%
LS-SVM linear kernel OvO	69.6%	5.6%	64.7%	5.8%	—	—	83.3%	4.6%	84.1%	3.9%
LS-SVM linear kernel OvA	40.9%	6.2%	44.5%	7.7%	—	—	63.2%	7.1%	54.8%	8.0%
LS-SVM RBF kernel OvO	74.4%	4.5%	69.9%	7.4%	—	—	84.1%	9.7%	88.9%	6.0%
LS-SVM RBF kernel OvA	62.5%	9.4%	48.8%	13.3%	—	—	72.3%	13.9%	80.9%	11.6%

therefore, very important to investigate the information provided to the classification models very thoroughly and exploit any available process knowledge in selecting the correct pretreatment for the fault classes of interest. Performance is also influenced by the type of classification model and its configuration, but these effects are generally less significant than the choice of pretreatment method.

Future research will consist of validating the conclusions on more complex case studies and measurement noise. In addition, the robustness of the classification models will be investigated by evaluating their performance for false alarms and for multiple faults occurring at the same time.

REFERENCES

Abe, S. (2005). *Support Vector Machines for Pattern Classification*. Springer-Verlag.

Aizerman, M.A., Braverman, E.A., and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Autom. Remote Control*, 25, 821–837.

Alcala, C.F. and Qin, S.J. (2011). Analysis and generalization of fault diagnosis methods for process monitoring. *J. Process Contr.*, 21(3), 322–330.

Birol, G., Ündey, C., and Çinar, A. (2002). A modular simulation package for fed-batch fermentation: penicillin production. *Comput. Chem. Eng.*, 26(11), 1553–1565.

Bodesheim, P., Freytag, A., Rodner, E., and Denzler, J. (2015). Local Novelty Detection in Multi-class Recognition Problems. In *Proc. 2015 WACV Conf.*, 813–820.

Boser, B.E., Guyon, I.M., and Vapnik, V.N. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proc. 5th Annual Workshop on Computational Learning Theory (COLT)*, 144–152.

Fix, E. and Hodges, J.L. (1951). Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical report 4, project 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas, USA.

Furnkranz, J. (2002). Round Robin Classification. *J. Mach. Learn. Res.*, 2, 721–747.

Gins, G., Van den Kerkhof, P., Vanlaer, J., and Van Impe, J. (2015). Improving classification-based diagnosis of batch processes through data selection and appropriate pretreatment. *J. Process Contr.*, 26, 90–101.

Gins, G., Vanlaer, J., Van den Kerkhof, P., and Van Impe, J. (2014). The RAYMOND simulation package – Generating RAYpresentative MONitoring Data to develop advanced process monitoring and control algorithms. *Comput. Chem. Eng.*, 69, 110–118.

Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling. *Ann. Stat.*, 451–471.

Jolliffe, I. (1986). *Principal component analysis*. Springer Verlag, New York.

King, R., Feng, C., and Sutherland, A. (1995). StatLog: Comparison of classification algorithms on large real-world problems. *Appl. Artif. Intell.*, 9(3), 289–333.

Li, B., Morris, J., and Martin, E. (2002). Model selection for partial least squares regression. *Chemometr. Intell. Lab.*, 64, 79–89.

MacGregor, J. and Kourti, T. (1995). Statistical Process Control of Multivariate Processes. *Control Eng. Pract.*, 3(3), 403–414.

Suykens, J., Van Gestel, T., and De Brabanter, J. (2002). *Least Squares Support Vector Machines*. World Scientific.

Ündey, C., Ertunç, S., and Çinar, A. (2003). Online Batch/Fed-Batch Process Performance Monitoring, Quality Prediction, and Variable-Contribution Analysis for Diagnosis. *Ind. Eng. Chem. Res.*, 42(20), 4645–4658.

Van den Kerkhof, P., Vanlaer, J., Gins, G., and Van Impe, J.F. (2013). Analysis of smearing-out in contribution plot based fault isolation for Statistical Process Control. *Chem. Eng. Sci.*, 104, 285–293.

Westerhuis, J.A., Gurden, S.P., and Smilde, A.K. (2000). Generalized contribution plots in multivariate statistical process monitoring. *Chemometr. Intell. Lab.*, 51(1), 95–114.

Wold, S., Kettaneh, N., Friden, H., and Holmberg, A. (1998). Modelling and diagnosis of batch processes and analogous kinetic experiments. *Chemometr. Intell. Lab.*, 44, 331–340.

Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z., Steinbach, M., Hand, D., and Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1), 1–37.

Ye, J. and Xiong, T. (2007). SVM versus Least Squares SVM. *J. Mach. Learn. Res. Proc.*, 644–651.

Zhao, C., Wang, F., Mao, Z., Lu, N., and Jia, M. (2008). Improved Batch Process Monitoring and Quality Prediction Based on Multiphase Statistical Analysis. *Ind. Eng. Chem. Res.*, 47(3), 835–849.