Computers and Chemical Engineering 194 (2025) 108918

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Smart Process Analytics for Process Monitoring

Fabian Mohr [a,1], Elia Arnese-Feffin [a,b,1], Massimliano Barolo [b], Richard D. Braatz [a,*]

[a] Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, 02139 Cambridge, MA, United States of America
[b] CAPE-Lab – Computer-Aided Process Engineering Laboratory, Department of Industrial Engineering, University of Padova, via Marzolo 9, 35131 Padova PD, Italy

ARTICLE INFO

Keywords:
Process analytics
Data analytics
Fault detection
Intelligent systems
Model identification

ABSTRACT

Process monitoring is critical to ensuring product quality and efficient, safe process operation. Data-driven modeling is used in the process industries to build fault detection systems. No single data-driven modeling method provides the best fault detection performance for all process systems, and the selection of the best data-driven modeling method for a specific process system requires substantial expertise. In this study, we propose Smart Process Analytics for Process Monitoring (SPAfPM), a systematic framework for automatic method selection and calibration of data-driven fault detection models. A set of candidate methods is pre-selected from a library on the basis of the characteristics of the data. A rigorous cross-validation procedure is then employed to compare the models obtained by these methods to select the best data-driven model for fault detection. The performance of SPAfPM is demonstrated in four case studies, including the Tennessee Eastman Process.

1. Introduction

Maintaining high product quality is a key requirement in most manufacturing processes, which can be achieved by process monitoring schemes. Fault detection is the first step in a chain of operations in process monitoring that are performed in order to recover a process to normal operating conditions in case any fault occurs (Fig. 1). After a fault is detected, the process/product quality variables most related to the malfunction are identified. The nature and, possibly, the root cause of the fault are then diagnosed leveraging expert process knowledge; alternatively, a classification approach can be used to diagnose which fault has actually occurred searching through a library of known faults. Finally, measures to recover the process operation are taken (Chiang et al., 2001). If those measures are not taken and the process is not recovered, the process can go out of control and lead to catastrophic events.

The terminology is not uniform in this area of research, and this article adopts the terminology and definitions of Raich and Çinar (1996). A fault is defined as "an unpermitted deviation of at least one characteristic property of a variable from an acceptable behavior" (Isermann, 2005). Such a deviation is considered a fault regardless of whether the deviation is caused by faulty equipment or a major disturbance. An example for faulty equipment is strong fouling in a heat exchanger that reduces the heat transfer rate required for process operation. Other examples of faulty equipment are biased sensors or clogged valves. An example of a significant disturbance is a raw material that is supplied from a different provider and has a high enough amount of impurities that the process is no longer able to produce material that meets quality specifications (Chiang et al., 2001).

Model-based process monitoring methods (Isermann, 1984, 1997) generally belong to one or more of three different categories: data-driven, analytical, and knowledge-based (Isermann, 1994). This work considers data-driven methods, which are most widely used in the materials, chemical, and biological industries. For fault detection, the first step is to construct a model that describes data from Normal Operating Conditions (NOC). Afterwards, statistical measures are used to decide whether new collected data deviate significantly from the data used to construct the NOC model (Qin, 2003). We refer interested readers to the literature for a comprehensive overview of process monitoring methods (Chiang et al., 2001; Qin, 2003; Venkatasubramanian et al., 2003c,a,b; Md Nor et al., 2020; Abid et al., 2021).

The data-driven approach requires the selection and calibration of a modeling method. However, numerous methods are available and no method performs best on all problems. Very few individuals possess significant expertise on all of the fault detection methods that can provide the best performance, and practitioners usually select the model to be used based on familiarity, even when the method is suboptimal for the particular application (Camacho and Ferrer, 2012; Camacho et al., 2009).

* Corresponding author.
E-mail address: braatz@mit.edu (R.D. Braatz).
[1] These authors contributed equally.

https://doi.org/10.1016/j.compchemeng.2024.108918
Received 4 April 2024; Received in revised form 7 October 2024; Accepted 5 November 2024
Available online 14 December 2024

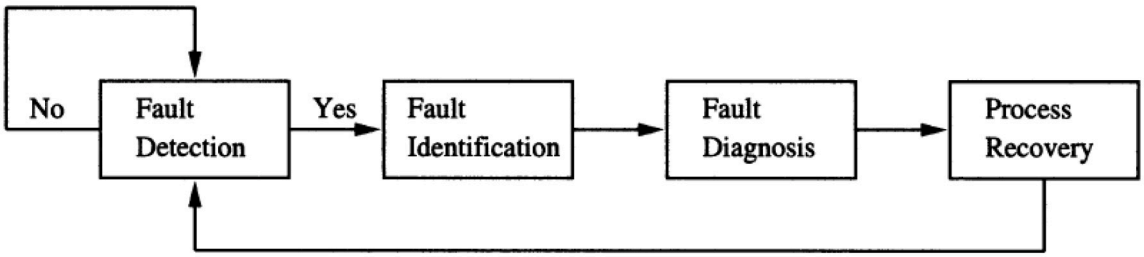

Fig. 1. The four steps involved in process monitoring (Chiang et al., 2001).

An alternative, more structured approach is to consider a set of candidate models and to select the best on the basis of the performance on data not used in calibration (i.e., in validation). Model selection and discrimination can be done on an independent validation dataset using the so-called hold-out validation (Bishop, 1995), or on the calibration dataset by means of resampling techniques, cross-validation (Stone, 1974; Allen, 1974) being the most popular choice. Comparing a large number of models on the basis of their performance in cross-validation is, in fact, the general principle underlying numerous frameworks for Automatic Machine Learning (AutoML; Hutter et al. (2019)). Some notable and recent packages include Auto-sklearn (Feurer et al., 2015), AutoWEKA (Kotthoff et al., 2017), Auto-Keras (Jin et al., 2019), TPOT (Le et al., 2020), H2OAutoML (H2O AI, 2024), TransmogrifAI (Salesforce, 2020), and MLJAR (MLJAR, 2024).

However, all the available AutoML packages are designed to handle supervised learning problems, mostly for making output predictions or classifications, while no automated system is available for building fault detection models. Furthermore, comparing a large number of candidate models by cross-validation has been proven to increase the chances of selecting a suboptimal model, especially when a limited amount of data is available (Arlot and Celisse, 2010). A further issue is that the cross-validation procedure is generally the same for all the models being compared, and their characteristics are disregarded, which is particularly relevant if models able to cope with different characteristics in the data, e.g., static vs. dynamic models, are compared. In fact, while cross-validation assumes that observations are independent (Arlot and Celisse, 2010), special procedures are required for dynamic data, where observations are autocorrelated (Bergmeir and Benítez, 2012). A final drawback of comparing multiple, possibly very different models on the basis of cross-validation alone lies in the fact that such a "winner takes all" approach disregards the appropriateness of the chosen model to the characteristics of the data at hand. As such, an inappropriate and non-robust model could show the best performance by chance and still be selected.

The aforementioned limitations have been discussed and illustrated by Sun and Braatz (2021). They proposed a bottom-up approach for automated method selection and calibration meant to tackle data-driven regression problems: Smart Process Analytics (SPA). The procedure starts with a preliminary assessment of the relevant properties of the data at hand (i.e., correlation, nonlinearity, and dynamics). Based on the outcome of the preliminary property assessment, methods that can cope by design with the detected characteristics are pre-select among the ones provided with the SPA method library. A rigorous cross-validation approach tailored to the characteristics of the selected methods is then used to identify the best model. The most relevant difference between AutoML packages and SPA lies in the additional pre-selection step, based on the characteristics of the data at hand: it ensures that only models able to cope with the data detected characteristics are compared by cross-validation, therefore effectively limiting the chances of overfitting.

In this article, we propose a SPA-like approach for automatizing the selection and application of the most suitable fault detection method for a given dataset. We demonstrate the approach, referred to as Smart Process Analytics for Process Monitoring (SPAfPM), in a number of case studies. Section 2 provides an overview of commonly used fault detection methods, their mathematical assumptions, and required characteristics. Section 3 describes the smart data analytics approach for fault detection, and Section 4 demonstrates the effectiveness of the approach on a variety of benchmark case studies, including the Tennessee Eastman Process (TEP).

2. Fault detection methods

SPAfPM provides a library of data-driven fault detection models. Dimensionality reduction methods based on variance modeling found several successful application in fault detection (Chiang et al., 2001), thus constitute the bulk of the models provided. The linear versions of such models are described in Section 2.1; dynamic and nonlinear extensions are introduced in Section 2.2. The data-driven fault detection problem can also be interpreted as an One-Class Classification (OCC; Brereton (2011)) task. We include an OCC model in SPAfPM and describe it in Section 2.3.

Note that we give brief descriptions of the rationale of the relevant methods here, while their mathematical details are reported in the Supplementary Material. A comprehensive overview of the methods discussed herein is given by Mohr (2024) and by Arnese Feffin (2023). We refer the reader to the literature cited throughout this Section and in the Supplementary Material for detailed descriptions of each method.

2.1. Linear dimensionality reduction methods

When a data matrix $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^m$ gathering N observations of m process (or input) variables is available, Principal Component Analysis (PCA) can be used for fault detection. PCA is a dimensionality reduction technique that captures the maximum variance of predictors in principal component vectors (e.g., Wold (1987), Chiang et al. (2001)). Specifically, the PCA model projects the data matrix onto a space of dimensionality $a \ll m$ defined by the principal components of matrix $\mathbf{X}$. As such, PCA "splits" the space of the process variables into the space of principal components (i.e., the model space) and into the so-called residual space (Ku et al., 1995).

If also a matrix $\mathbf{Y} \in \mathbb{R}^N \times \mathbb{R}^l$ gathering N observations and l output variables (e.g., characterizing the product quality) is given, the Partial Least-Squares (PLS) regression model can be used for fault detection. PLS is a dimensionality reduction technique that maximizes the covariance between the input variables and the output variables for each component of the reduced space (e.g., Geladi and Kowalski (1986), Wold et al. (2001), Jiao et al. (2015), Chiang et al. (2001)). Similarly to PCA, the data matrices are projected onto spaces with the same dimensionality a defined by two sequences of latent variables. Couples of input and out latent variables are computed in such a way to maximize the linear correlation among them (while at the same time retaining as much variances of the input and output spaces as possible). In other words, only the process variability affecting the product quality is modeled. This approach justifies the use of PLS for quality-relevant

monitoring, i.e., to detect only process faults affecting the product quality.

Canonical Variate Analysis (CVA) operates similarly to PLS, but also includes information on the dynamic evolution of the variables in matrices $\mathbf{X}$ and $\mathbf{Y}$ (e.g., Larimore (1990), Russell et al. (2000), Chiang et al. (2001), Jiang et al. (2015)). The observations in the data matrices are interpreted as realizations of random processes and used to construct a past and future matrices. The past matrix $\mathcal{P} \in \mathbb{R}^{N-h-j} \times \mathbb{R}^{(m+l)h}$ gathers lagged observations h (prior to the one at the current time) of both the input and output variables. The future matrix $\mathcal{F} \in \mathbb{R}^{N-h-j} \times \mathbb{R}^{l(1+j)}$ collects output observations at the current time and at j time future instants. CVA projects both matrices onto a common space of reduced dimensionality maximizing the correlation between two sets of variables, which are the past and future vectors (i.e., rows of matrices $\mathcal{P}$ and $\mathcal{F}$, respectively). Therefore, CVA can be interpreted as a state-space modeling method, and the space identified by CVA as the state-space of the process.

PCA, PLS, and CVA share a common feature: variables are projected onto a model space, while the unmodeled part is left in the residual space. Variations of data within these spaces can be monitored with the T^2 and Q statistics (Wold, 1987; Nomikos and MacGregor, 1995; Chiang et al., 2001; Qin, 2003), respectively. Specifically, the T^2 statistic describes the squared distance of an observation from the center of the model space, while the Q statistic quantifies the squared orthogonal distance of an observation from the model space itself. The CVA model provides and additional statistic, i.e., T_r^2, which measures the variability of an observation outside of the model space (Russell et al., 2000).

Faults can be detected by comparing the values of the aforementioned statistics to some control limits, the values of which can be defined in a number of ways (e.g., Reis et al. (2021)). In SPAfPM, the control limit of the T^2 statistic (and of the T_r^2 statistic) can be estimated using the F distribution approach (Jackson, 1959) or the χ^2 distribution approach (Nomikos and MacGregor, 1995). The control limit of the Q statistic can be estimated using the Jackson-Mudholkar method (Jackson and Mudholkar, 1979) or the χ^2 distribution approach (Nomikos and MacGregor, 1995). The formulations of the control limits are reported in the Supplementary Material.

2.2. Dynamic and nonlinear transformations

The aforementioned methods can model only static correlation among variables, with the exception of CVA. Furthermore, only linear relationships can be modeled using PCA, PLS, and CVA. In this Section, extensions of the basic algorithms to the dynamic and nonlinear cases (and their combination) are briefly discussed.

A number of dynamic extensions of the basic PCA and PLS algorithms exist, which are usually referred to as Dynamic Principal Component Analysis (DPCA; Ku et al. (1995)) and Dynamic Partial Least-Squares (DPLS; Ricker (1988)). These methods are based on the idea of lagged observations already discussed for CVA. The data matrix $\mathbf{X}$ is typically augmented with additional variables given by h past observations. This results in the so-called trajectory matrix $\mathbf{X}_h \in \mathbb{R}^{N-h} \times \mathbb{R}^{m(1+h)}$, which serves as the basis of DPCA and DPLS.

DPCA aims to model autocorrelation and cross-correlation in the dataset, implicitly extracting a dynamic autoregressive model of the process (Ku et al., 1995). DPCA is performed by applying the regular PCA algorithm to matrix $\mathbf{X}_h$. Regarding DPLS, the most commonly used approaches applies the regular PLS algorithm using $\mathbf{X}_h$ as input matrix and leaving unaltered the output matrix $\mathbf{Y}$ (removing the first h rows to even out the number of observations). This approach incorporates dynamics in PLS by the same rationale of a finite impulse response model (Ricker, 1988; Jiao et al., 2015; Jia and Zhang, 2016).

DPCA and DPLS still use the T^2 and Q, which can be estimated in the same way as for the standard PCA and PLS. However, dealing with dynamic data requires additional considerations when estimating the control limits. The χ^2 distribution approach is recommended when applying dynamic extensions such as DPCA and DPLS (Lu et al., 2005; Yao and Gao, 2007).

Dynamic extension of PCA and PLS are obtained manipulating the input data matrix. A similar principle yields nonlinear extensions of the regular algorithms. Specifically, applying kernel transformations to the input matrix yields Kernel Principal Component Analysis (KPCA; Schölkopf et al. (1998)) and Kernel Partial Least-Squares (KPLS; Rosipal and Trejo (2001)).

Assuming that the variables feature nonlinear correlation, the fundamental idea of kernel methods is to project the observations onto a high-dimensional space, called the feature space, by means of nonlinear transformations (Müller et al., 2001). The mapping function is defined in such a way that the relationship among transformed variables is linear in the feature space, thus it can be modeled using the regular PCA and PLS models (Schölkopf et al., 1998; Rosipal and Trejo, 2001). To avoid an explicit mapping, which could result in a computationally infeasible problem, KPCA and KPLS exploit the kernel trick: a pairwise kernel function $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is applied to each pair of observations in the input matrix to compute the so-called kernel matrix $\mathbf{K} \in \mathbb{R}^N \times \mathbb{R}^N$; $K(\mathbf{x}_i, \mathbf{x}_j)$ yields the entry on the ith row and jth column of $\mathbf{K}$. Popular kernel function are the Radial Basis Function (RBF), also known as the Gaussian kernel, and the polynomial kernels.

KPCA (Schölkopf et al., 1998) is obtained by applying regular PCA in the feature space. Under the kernel trick, the regular PCA algorithm is applied to the kernel matrix; the method includes some tailored pre- and post-processing operations to account for the implicit transformation operated by the pairwise kernel function. KPLS (Rosipal and Trejo, 2001) works in a similar manner: regular PLS is applied using matrix $\mathbf{K}$ as input and an unchanged matrix $\mathbf{Y}$ as output (and considering some tailored pre- and post-processing operations).

Nonlinear transformation of the data matrices can also be combined with the augmentation by lagged observations: this is the principle of Dynamic Kernel Principal Component Analysis (DKPCA; Choi and Lee (2004)) and Dynamic Kernel Partial Least-Squares (DKPLS; Jia and Zhang (2016)). In this case, instead of applying the kernel transformations to the inputs directly, they are applied to the trajectory matrix including lags of the basic variables $\mathbf{X}_h$. In this way, both dynamic and nonlinear effects in the data can be considered (Choi and Lee, 2004; Jia and Zhang, 2016).

For both KPCA and KPLS (and their dynamic versions DKPCA and DKPLS), the T^2 and the Q statistic can be computed in the same way as for the regular PCA and PLS. The T^2 statistic is still related to the model space, while the Q statistic is related to the residual space transformed by the kernel function (Choi et al., 2005; Cho et al., 2005; Zhang and Qin, 2008). The control limit estimators remain unchanged.

Finally, nonlinear extensions of CVA have also been developed. One such extension (Odiowei and Cao, 2010) is based on the regular CVA algorithm to model the dynamics in the data; nonlinearity is accounted for at the fault detection statistics level, whose control limits are estimated by Kernel Density Estimation (KDE; Rosenblatt (1956), Parzen (1962)). This method is referred to as KDE-CVA (Odiowei and Cao, 2010). See the Supplementary Material for details.

2.3. Support vector data description

Other modeling paradigms can be fruitfully exploited for process monitoring tasks. OCC (Brereton, 2011) is a noteworthy example. OCC tackles the fault detection problem by constructing a description of data coming from a single class (i.e., the NOC data); this allows to determine whether new observations conform to the characteristic of the modeled class or not (Rodionova et al., 2016; Tax and Duin, 1999, 2004). OCC models can be used to detect observations that significantly differ from the modeled class (Tax and Duin, 1999) and new data conditions (Rodionova et al., 2016), e.g., faults. Estimating the support of the distribution of the modeled class is a popular approach to OCC

(Müller et al., 2001), which involves modeling the boundaries of the class. This procedure is appropriate when no *a priori* assumption can be done about the distribution of the out-of-class (i.e., faulty) data (Tax and Duin, 1999).

Support Vector Data Description (SVDD; Tax and Duin (1999, 2004)) is an OCC method based on the concept of distribution support modeling. In its linear version, SVDD identifies a hypersphere of minimal radius that encloses all observations in a given data matrix **X**; the algorithm is designed to allows some observations to lie outside of the hypersphere to deal with possible outliers in the dataset. Once model calibration is complete, SVDD yields the center of the hypersphere, expressed as a linear combination of some observations known as support vectors, and the radius of the hypersphere. These entities can be used to test whether a new observation falls within the hypersphere (i.e., it is NOC) or outside of it (i.e., it is a not NOC). Therefore, the distance D of a new observation from the center of the hypersphere serves as the fault detection statistic, and the radius R of the hypersphere is the control limit. In SPAfPM, the nonlinear version of SVDD is considered, which combines the kernel transformation discussed in the previous Section with the SVDD algorithm to model the boundary of complex (i.e., non-normal/nonlinear) distributions. See the Supplementary Material for details.

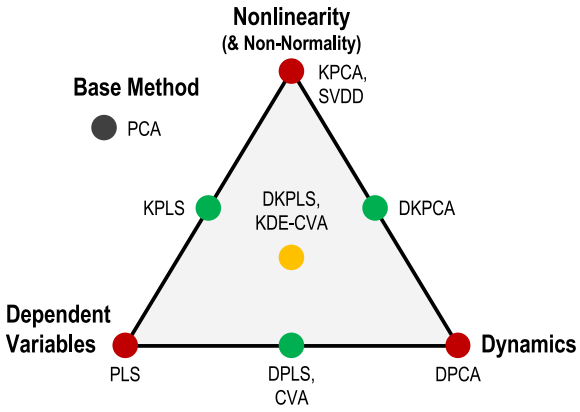

Fig. 2. The smart data analytics triangle for fault detection. PCA: Principal Component Analysis, DPCA: Dynamic PCA, KPCA: Kernel PCA, DKPCA: Dynamic Kernel PCA, PLS: Partial Least-Squares, DPLS: Dynamic PLS, KPLS: Kernel PLS, DKPLS: Dynamic Kernel PLS, CVA: Canonical Variate Analysis, KDE-CVA: Kernel Density Estimation-CVA, SVDD: Support Vector Data Description.

3. A smart data analytics approach to fault detection

The performance of each algorithm reviewed in Section 2 can widely vary when applied to different datasets due to the underlying assumptions of methods and of their match to the characteristics of the data. Ultimately, such the data characteristics determine which method is most appropriate for a given dataset. Therefore, they must be considered in the development of data-driven process monitoring systems.

Some fundamental data characteristics can be identified: nonlinearity of the relationships among variables (or non-normality of the data distribution), dynamics in process variables, and presence of variables to characterize the product quality. We use these characteristics as foundation for SPAfPM to select the best model for the data at hand. We choose these characteristics as they are very common in data from industrial processes. We discuss the data characteristics and how they can the pre-selection of appropriate models in Section 3.1, while we develop and evaluate automated test to detect the characteristics in Section 3.2. Finally, we discuss the model selection mechanisms of SPAfPM in Section 3.3.

3.1. Data analytics triangle for fault detection

Similarly to other smart data analytics approaches (Sun and Braatz, 2021; Mohr et al., 2022), a base method for the given task (i.e., fault detection) can be identified. The base method chosen for the proposed framework is PCA by virtue of its wide usage in the process monitoring literature and proven performance in fault detection. PCA can cope with large datasets containing correlated variables, a trait that is reasonable to expect in data commonly used for process monitoring, often including measurements of all available process variables (Wise and Gallagher, 1996). PCA relies on three assumptions:

- correlation among variables is linear (Wold, 1987);
- data follow a multivariate normal distribution (for the reliability of control limits of monitoring statistics) (Qin, 2003);
- no dynamics are found in the data and/or residuals (Ku et al., 1995).

As such, PCA is appropriate when the data at hand do not posses any of the characteristics mentioned in the introduction to Section 3. This further justifies its choice as the base method in SPAfPM.

The assumptions of linear correlation and absence of dynamics are required due to the PCA working principle that defines latent variables as static, linear combinations of observable variables. On the other hand, the normality assumption is required to ensure the reliability of the monitoring statistics. In fact, the matrix decomposition of PCA is based on the covariance matrix of data, second-order information that is enough to describe only multivariate normal distributions. Furthermore, control limits of the monitoring statistics as reviewed in Section S.1.1 (Hotelling's T^2 and Q) are fully descriptive only under the assumption that scores and residuals are normally distributed (Thissen et al., 2001). The scores are normally distributed only if the input data are normally distributed as a linear combination of normal variables is still normal (Nomikos and MacGregor, 1994). On the other hand, residuals are normally distributed only if all the systematic variability (including the potential dynamics) has been captured by the model and transferred to the latent space (Wold, 1987).

We argue that non-normality and nonlinearity are tightly intertwined characteristics of a dataset (we provide a brief discussion on this matter in Section 3.2.2 and further investigate this hypothesis in Section 3.2.2). Methods able to cope with nonlinear correlation based on kernel transformations, such as KPCA, can deal by design with general (i.e., non-normal) distributions. Therefore, non-normality and nonlinearity are checked independently in SPAfPM, but are considered as a single characteristic of data. The presence of dynamics in the data is another characteristic that is assessed.

Data from process industrial processes frequently include discrete variables, (e.g., to indicate the phase of a recipe-driven batch process). If such variables, are present, particular attention is due when assessing data characteristics such as nonlinearity and dynamics. We provide a brief discussion of this topic in the Supplementary Material, and consider a case study involving discrete variables in Section 4.4.

A decision to be made is whether the objective is monitoring of the *overall process* or just the variability affecting *the quality of the final product*, therefore whether to adopt a "general" monitoring scheme or a quality-relevant monitoring approach (Li et al., 2011). While the first aim can be achieved with PCA, the second aim (provided that online quality measurements are available) can be achieved with PLS, which relies on similar assumptions as for PCA concerning the extraction of latent variables. Therefore, a third characteristic of the data is the presence of quality variables so as to develop quality-relevant monitoring systems. The identification of quality variables requires expert knowledge from the users of the smart data analytics software. Consequently, the designation of dependent variables is a choice left to the user.

The data characteristics and relevant associated data analytics methods in the proposed framework can be visualized in the form of a smart

Table 1
Overview of the different hyperparameters considered for cross-validation.

Algorithm	Hyperparameter	Meaning	Notes
PCA	a	Number of principal components	
PLS	a	Number of latent variables	
	h	Extent of past horizon	
CVA	j	Extent of future horizon	$j = h$
	a	State order	
	C	Coverage parameter	
SVDD	$K(\cdot,\cdot)$	Pairwise kernel function	
	σ or (c_0, d, γ)	Kernel parameter (RBF or poly)	$c_0 = 1$
	h	Number of lags	
DPCA	a	Number of principal components	
	$K(\cdot,\cdot)$	Pairwise kernel function	
KPCA	σ or (c_0, d, γ)	Kernel parameter (RBF or poly)	$c_0 = 1$
	a	Number of principal components	
	h	Number of lags	
	$K(\cdot,\cdot)$	Pairwise kernel function	
DKPCA	σ or (c_0, d, γ)	Kernel parameter (RBF or poly)	$c_0 = 1$
	a	Number of principal components	
DPLS	h	Number of lags	
	a	Number of latent variables	
	$K(\cdot,\cdot)$	Pairwise kernel function	
KPLS	σ or (c_0, d, γ)	Kernel parameter (RBF or poly)	$c_0 = 1$
	a	Number of latent variables	
	h	Number of lags	
DKPLS	$K(\cdot,\cdot)$	Pairwise kernel function	
	σ or (c_0, d, γ)	Kernel parameter (RBF or poly)	$c_0 = 1$
	a	Number of latent variables	
	h	Number of past lags	
KDE-CVA	j	Extent of future horizon	$j = h$
	a	Memory order	
	ξ_{T^2}	Scale factor for kernel bandwidth of T^2	
	ξ_Q	Scale factor for kernel bandwidth of Q	
	$\xi_{T_r^2}$	Scale factor for kernel bandwidth of T_r^2	$\xi_{T_r^2} = \xi_{T^2}$

data analytics triangle for fault detection (Fig. 2). The triangle is built around the three aforementioned core characteristics of the available dataset: the presence of dependent variables in the data, nonlinearity or non-normality, and dynamics. If none of the characteristics is detected, the base method, PCA, is applied. The corners represent algorithms to apply for one of the characteristics present. The edges show the fault detection algorithms suitable for the characteristics at the linked corners. The center of the triangle shows algorithms best suited if all three characteristics are present in the dataset.

For example, if the data feature nonlinearity and dependent variables, the data triangle suggests to use the KPLS algorithm. Only one algorithm is suggested in this case. However, if the data feature dynamics and dependent variables, two different algorithms are recommended: DPLS and CVA. In this case, a cross-validation procedure is applied to determine which of the two algorithms is best for the given case and to determine the optimal hyperparameters. An overview of the different hyperparameters for each one of the algorithms in the data analytics triangle is shown in Table 1. The structure of the cross-validation procedure is explained in detail in Section 3.3. A schematic highlighting the overall workflow in conjunction with the smart data analytics triangle is visualized in Fig. 3.

In the following Sections, a data interrogation framework for the characteristics non-normality/nonlinearity and dynamics is presented and demonstrated in rigorous Monte Carlo simulations. Based on the detected characteristics, the best fault detection algorithm can be selected from the presented triangle. Additionally, the cross-validation procedure is described in detail.

3.2. Preliminary data interrogation

The quantitative criteria used to assess the relevant data characteristic introduced in the previous section, i.e., non-normality, nonlinearity, and dynamics, are introduced in this section. The effectiveness of the criteria is demonstrated using rigorous Monte Carlo simulations. All the simulations are carried out using Python, version 3.9.12 and R, version 4.2.0.

3.2.1. Non-normality detection

Mecklin and Mundfrom (2005) carried out a Monte Carlo study on the effectiveness of various multivariate normality tests that indicated that the Henze-Zirkler test (Henze and Zirkler, 1990) is preferred due to its better empirical performance and theoretical properties. Royston's test (Royston, 1983) performed very well nonetheless, to the level of the Henze-Zirkler test (Mecklin and Mundfrom, 2005). Mardia's skewness and kurtosis tests (Mardia, 1970) showed good performance as well, and are among the most widely used tests for multivariate normality. For a concise overview of the mathematical formulation of the aforementioned tests, see Korkmaz et al. (2014).

Preliminary analyses on the four mentioned tests highlighted pros and cons of each. Considerations regarding the theory of the tests further backed up the empirical results. The most important points are:

- All the aforementioned tests (i.e., the Henze-Zirkler test, Royston's test, and Mardia's test) require the inversion of the covariance matrix of the sample, therefore cannot be applied for a singular sample correlation matrix (e.g., if there are more variables than observations).
- Royston's test can be applied to samples with up to 2000 observations due to its formulation.
- The statistic used in the Henze-Zirkler test is based on the lognormal distribution and its variance shrinks to zero as the number of variables increases (unless balanced by a remarkably large number of observations). The propagation of numerical errors become increasingly important with increasing number of variables, which compromises the reliability of the test. We briefly discuss this effect in this Section and provide additional details in the Supplementary Material.

Given these preliminary considerations, a Monte Carlo study is carried out to properly evaluate performances of the four tests. The factors considered in the Monte Carlo studies are:

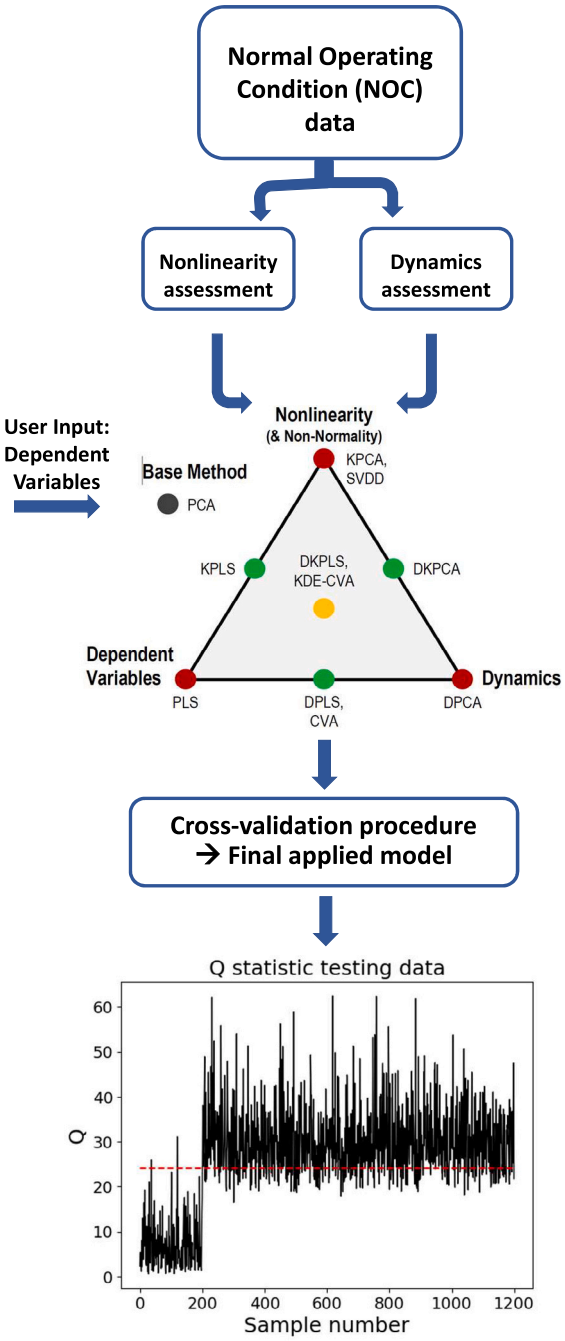

Fig. 3. Visualization of overall approach. PCA: Principal Component Analysis, DPCA: Dynamic PCA, KPCA: Kernel PCA, DKPCA: Dynamic Kernel PCA, PLS: Partial Least-Squares, DPLS: Dynamic PLS, KPLS: Kernel PLS, DKPLS: Dynamic Kernel PLS, CVA: Canonical Variate Analysis, KDE-CVA: Kernel Density Estimation-CVA, SVDD: Support Vector Data Description.

- The distribution used to generate the sample, D, which can be multivariate normal, multivariate t, multivariate lognormal, or multivariate uniform.
- The number of variables in the sample: $m \in \{10, 30, 50, 100, 200\}$.
- The number of observations in the sample: $N \in \{50, 200, 500, 1000, 3000\}$.

All possible combinations of the factors are explored; however, combinations where the number of observations is greater than the number of variables are not explored. For each combination, 100 repetitions are performed. For each repetition, a sample is generated from the selected distribution using randomly selected parameters (different at each repetition).

The four normality tests (Henze-Zirkler, Royston's, Mardia's skewness, and Mardia's kurtosis tests) are performed on the generated sample. The outcomes of the four tests are saved for each repetition for a given combination of factors and used to compute the non-normality detection rates (i.e., the number of repetitions over which the sample is deemed non-normal over the total number of repetitions) of each one of the four tests. The four non-normality detection rates are the responses of the Monte Carlo study. These detection rates should be as close as possible to the chosen significance level ($\alpha = 0.01$) for the multivariate normal distribution, and to its complementary to one ($\beta = 0.99$) for all the other distributions.

The results of the four selected tests are further combined in order to yield two more responses for the Monte Carlo study, which are also reported in the below discussion. The two additional responses are:

- Results from Mardia's skewness and kurtosis tests are used to obtain the detection rate of Mardia's combined test (a dataset is deemed non-normal if either one of the two tests detects non-normality).
- Results from the four tests are combined in the "overall" test described at the end of this Section.

A second Monte Carlo study is set up, modifying the sample generation mechanism. The "sampling distribution" factor is replaced with the "fraction of nonlinear variables" factor. The domain of such a factor is: $f_{\text{nl}} \in \{0, 0.05, 0.10, 0.20, 0.40, 0.80\}$. To better understand how the sample is generated, assume, for example, that $m = 25$, and that 30 % of the variables are nonlinearly correlated ($f_{\text{nl}} = 0.3$) with the remaining 70 % of variables, which can feature a varying degree of linear correlation among one another. The first step is to sample $m_{\text{lin}} = \lfloor 0.7m \rfloor = 17$ variables from a multivariate normal distribution with randomly generated parameters. Then, $m_{\text{nl}} = m - m_{\text{lin}} = 8$ additional variables are generated by randomly picking m_{nl} out of the m_{lin} linear variables (with replacement, if $m_{\text{nl}} > m_{\text{lin}}$) and applying nonlinear transformations randomly selected from a library of sixty nonlinear transformations. White noise is added to each one of the m_{nl} nonlinear variables sampling normal distributions with zero mean and variance such that the signal-to-noise ratio of the transformed variables is 1:0.1. Finally, the m_{lin} linear variables and the m_{nl} nonlinear variables are jointed to produce the sample. Responses of the second Monte Carlo study are the non-normality detection rates of the six aforementioned tests.

Results of the Monte Carlo simulations on detection of normality are shown in Fig. 4. Royston's test performed the best overall, always yielding non-normality detection rates very close to the nominal significance level. The Henze-Zirkler test was nearly equivalent in terms of performance for most cases. Performance visibly deteriorated, however, when the sample includes more than 50 variables (non-normality is detected by default as the test statistics is stuck to its maximum value, which causes the p-value to be always 0). Such behavior is due to the aforementioned variance shrinkage of the lognormal distribution used to compute the test statistic. See the Supplementary Material for additional details. Mardia's skewness test also performed well, but Mardia's kurtosis test did not perform as well due to the inherent difficulty in properly characterizing the kurtosis of high-dimensional multivariate distributions.

Considering results on other distributions (see the Supplementary Material for detailed results), we can draw some conclusions:

- All tests yielded nearly the same performance when applied to the multivariate lognormal distribution, which is highly non-normal.
- All tests yielded nearly the same performance when applied to the multivariate t distribution, which is slightly non-normal and converges to a multivariate normal distribution for increasing degrees of freedom. The Henze-Zirkler test performed marginally better than others for small sample sizes, although also exhibiting more erratic results.

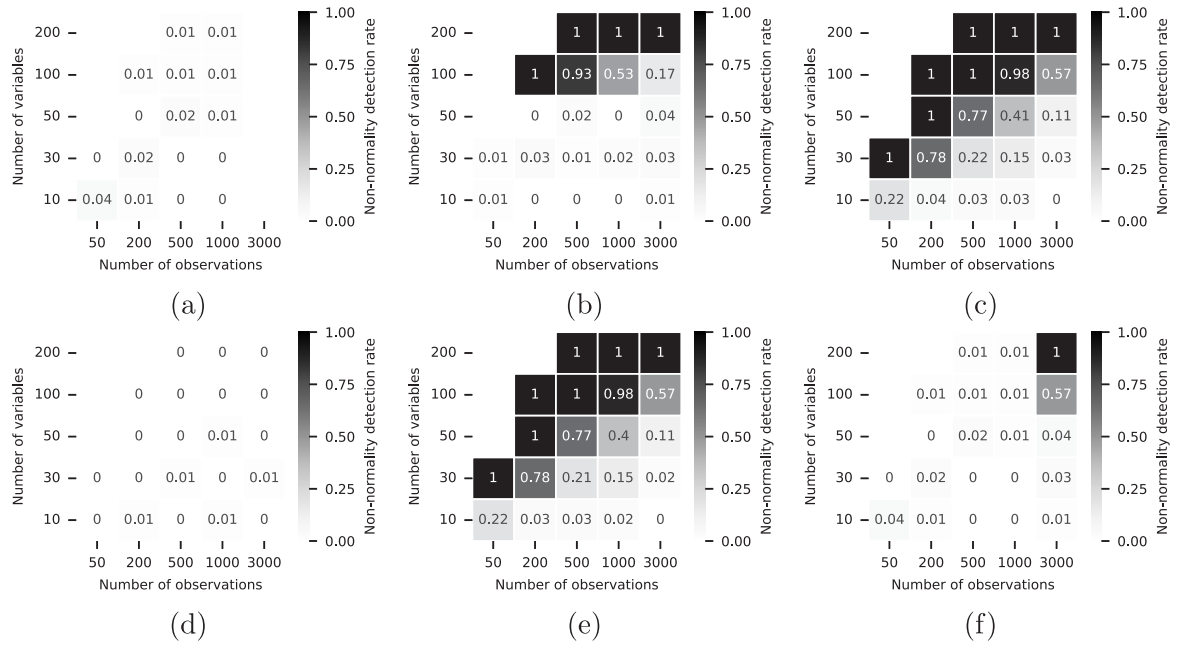

Fig. 4. Non-normality detection rates of multivariate normality tests on samples generated from multivariate normal distributions: (a) Royston's test, (b) Henze-Zirkler test, (c) Mardia's combined test, (d) Mardia's skewness test, (e) Mardia's kurtosis test, and (f) combination of all of the tests. Missing values mean that the relevant tests are not applicable for a given combination of factors.

- Royston's test outperformed other tests on the multivariate uniform distribution. In particular, the Henze-Zirkler test yielded very erratic results in this case, even for less than 50 variables.

All these observations are also seen in the second Monte Carlo study, in which the number of nonlinear variables is manipulated rather than the whole distribution. Royston's test performed slightly better than Henze-Zirkler test for mild deviations from normality ($f_{\text{nl}} = 0.05$ and $f_{\text{nl}} = 0.1$), especially for small sample sizes. In this case, both Mardia's tests yielded erratic results, as in the case of $f_{\text{nl}} = 0.2$ shown in Fig. 5, which is also the case where Royston's test outperformed Henze-Zirkler test most apparently, the latter exhibiting very erratic results. The performance of all tests converge for high fractions of nonlinear variables, where deviations from normality become apparent. Mild deviations from normality are hard to detect, as expected, especially on small samples (see the Supplementary Material for detailed results).

In light of these observations and bearing in mind remarks made by Mecklin and Mundfrom (2005), the default criterion to test non-normality of the dataset is selected as Royston's test, being the test that offers the best balance between performance and range of applicability. If the sample includes more than 2000 observations, the Henze-Zirkler test is used when there are at most 50 variables, and the combined Mardia's test is used otherwise. We provide additional insight on the reason to choose $m = 50$ as threshold for test switching in the Supplementary Material.

3.2.2. Nonlinearity detection

An assumption underlying PCA (and PLS) is that only linear correlations are in the data (Wold, 1987; Camacho et al., 2008). Also assumed is that the data are normally distributed, which implies that the noise is completely described by second-order statistics. As such, unsatisfactory monitoring performance has been reported when PCA is applied to non-normal data (Zhu et al., 2016). On the other hand, the presence of nonlinear correlation among variables implies that data do not follow a normal distribution (see Section 3.2.2).

The proposed nonlinearity detection method is based on three tests performed simultaneously: linear correlation analysis (Montgomery and Runger, 2018), maximal correlation analysis (Rényi, 1959) by the alternating conditional expectation algorithm (Breiman and Friedman, 1985), and quadratic (correlation) test (Montgomery and Runger, 2018) with adjustment of the significance level by the Bonferroni correction (Hochberg, 1988).

Given two samples $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ of the random variables X and Y, the sample linear correlation coefficient (Montgomery and Runger, 2018) can be computed as

$$r_{\mathbf{x},\mathbf{y}} = \frac{s_{\mathbf{x},\mathbf{y}}}{s_{\mathbf{x}} s_{\mathbf{y}}} = \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_i - \bar{x})^2}\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{1}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of X and Y, respectively, $s_{\mathbf{x}}$ and $s_{\mathbf{y}}$ are their sample standard deviations, and $s_{\mathbf{x},\mathbf{y}}$ is the sample covariance between X and Y. The linear correlation coefficient quantifies the degree of linear correlation between the two variables and varies between -1 and 1. Variables are uncorrelated if $r_{\mathbf{x},\mathbf{y}} \simeq 0$, while they are perfectly (anti-) correlated if $r_{\mathbf{x},\mathbf{y}} \simeq 1$ ($r_{\mathbf{x},\mathbf{y}} \simeq -1$).

The sample maximal correlation coefficient is defined as (Rényi, 1959)

$$r^*_{\mathbf{x},\mathbf{y}} = \sup_{\theta,\phi}[r_{\theta(\mathbf{x}),\phi(\mathbf{x})}] \tag{2}$$

where θ and ϕ are functions from the set of all the measurable Borel functions with zero mean, and are applied element-wise to $\mathbf{x}$ and $\mathbf{y}$. The sample maximal correlation coefficient can be computed by means of the alternating conditional expectation algorithm, which is suitable to deal with discrete variables (for instance, categorical or binary) by default (Breiman and Friedman, 1985). The maximal correlation coefficient domain is $r^*_{\mathbf{x},\mathbf{y}} \in [0, 1]$, where the transformed variables $\theta(X)$ and $\phi(Y)$ are uncorrelated if $r^*_{\mathbf{x},\mathbf{y}} \simeq 0$, and perfectly correlated if $r^*_{\mathbf{x},\mathbf{y}} \simeq 1$. Comparing the absolute value of the linear correlation coefficient and the value of the maximal correlation coefficient provides an understanding of the nature of the relationship between X and Y:

- If $r_{\mathbf{x},\mathbf{y}} \simeq 0$ and $r^*_{\mathbf{x},\mathbf{y}} \simeq 0$, the variables are uncorrelated.
- If $r_{\mathbf{x},\mathbf{y}} \simeq 1$ and $r^*_{\mathbf{x},\mathbf{y}} \simeq 1$, the variables are linearly correlated (the functions θ and ϕ are the identity functions).
- If $r_{\mathbf{x},\mathbf{y}} \simeq 0$ and $r^*_{\mathbf{x},\mathbf{y}} \simeq 1$, the variables are nonlinearly correlated.

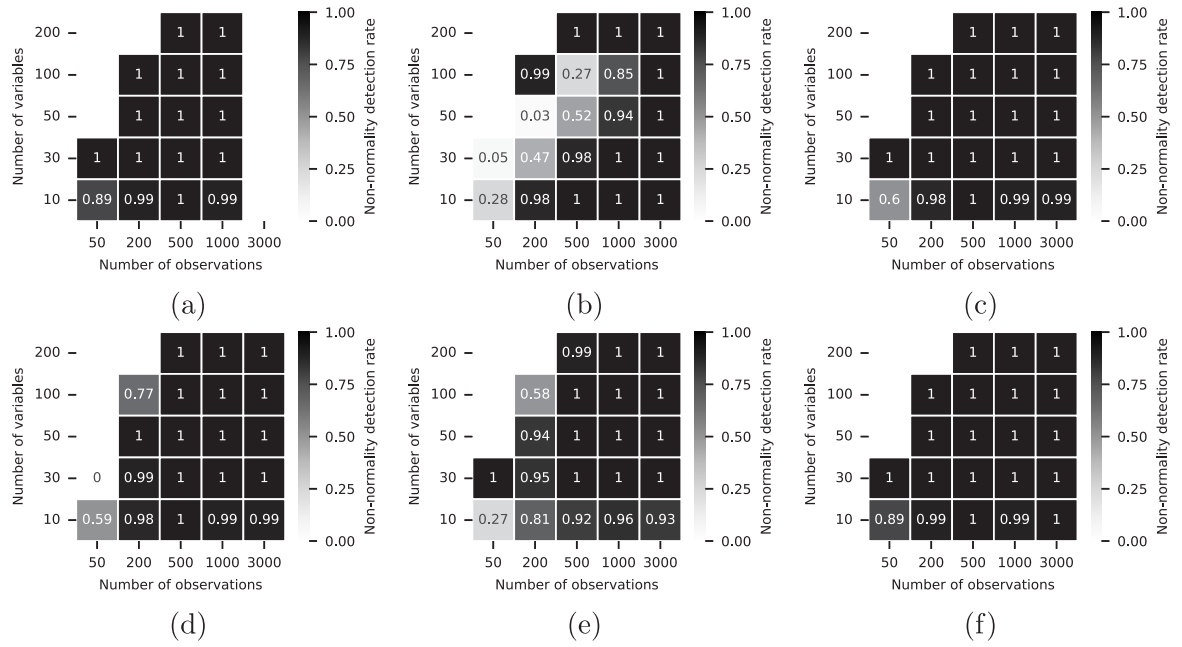

Fig. 5. Non-normality detection rates of multivariate normality tests on samples in which 20 % of the variables are nonlinear. The tests are in the same order as in Fig. 4. Missing values mean that the relevant tests are not applicable for a given combination of factors.

The quadratic test (Montgomery and Runger, 2018) is based on the idea of assessing the presence of a quadratic relationship between the random variables X and Y by first fitting a linear regression model and a quadratic regression model to given samples, then comparing the performances of the two by means of the analysis of variance. The null hypothesis is that the relationship is linear, while the alternative hypothesis is that the relationship is quadratic. The hypotheses are formulated as

$$H_0 : \mathbf{y} = b_{1_0}\mathbf{x} + b_{0_0} + \boldsymbol{\varepsilon}_0, \tag{3}$$

$$H_a : \mathbf{y} = b_{2_a}\mathbf{x}^2 + b_{1_a}\mathbf{x} + b_{0_a} + \boldsymbol{\varepsilon}_a, \tag{4}$$

where $\boldsymbol{\varepsilon}_0 \in \mathbb{R}^N$ and $\boldsymbol{\varepsilon}_a \in \mathbb{R}^N$ are samples of normal random variables. The F-test can be applied for hypothesis testing, with the F-value computed from

$$F_{\text{val}} = \frac{\frac{\text{MSE}_0 - \text{MSE}_a}{\text{DF}_0 - \text{DF}_a}}{\frac{\text{MSE}_a}{\text{DF}_a}}, \tag{5}$$

where MSE_0 and MSE_a are the mean squared errors of the linear and quadratic models, respectively, and DF_0 and DF_a are their numbers of degrees of freedom. The test statistic is distributed as an F variable with $\text{DF}_0 - \text{DF}_a$ numerator degrees of freedom and DF_a denominator degrees of freedom, so the p-value associated to the F-value can be computed from

$$p_{\text{val}} = 1 - F(\text{DF}_0 - \text{DF}_a, \text{DF}_a)|_{F_{\text{val}}}, \tag{6}$$

where $F(\text{DF}_0 - \text{DF}_a, \text{DF}_a)|_{F_{\text{val}}}$ is the value of the inverse cumulative distribution function of the F variable evaluated at F_{val}. The p-value can then be compared to the significance level of the test adjusted by the Bonferroni correction (Hochberg, 1988), α_{QTadj}. The quadratic correlation is deemed significant if $p_{\text{val}} < \alpha_{\text{QTadj}}$. The correction of the significance level is employed to control the false-positive rate when a large number of tests is performed simultaneously (Nadon and Shoemaker, 2002; Goeman and Solari, 2014).

The nonlinearity assessment method used in our framework is based on the method originally proposed for SPA (Sun and Braatz, 2021). The nonlinear correlation between a pair of variables is deemed significant if at least one of the following two conditions is verified:

- Linear correlation coefficient is close to 0, while maximal correlation coefficient is close to 1.
- The p-value of the quadratic test is below the adjusted threshold.

As such, two tests are conducted.

The first test regards the significance of the difference between the maximal correlation coefficient and the absolute linear correlation coefficient. The test is based on two conditions:

- If $r^*_{\mathbf{x},\mathbf{y}} \leq \varepsilon_{\text{MC}}$, the nonlinear correlation is deemed significant if $(r^*_{\mathbf{x},\mathbf{y}} - D) - \left|r_{\mathbf{x},\mathbf{y}}\right| > \varepsilon_1$, and as insignificant otherwise.
- If $r^*_{\mathbf{x},\mathbf{y}} > \varepsilon_{\text{MC}}$, the nonlinear correlation is deemed significant if $r^*_{\mathbf{x},\mathbf{y}} - \left|r_{\mathbf{x},\mathbf{y}}\right| > \varepsilon_2$, and as insignificant otherwise.

Default values of thresholds are set as in SPA (Sun, 2020a): $\varepsilon_{\text{MC}} = 0.92$, $\varepsilon_1 = 0.40$, and $\varepsilon_2 = 0.10$. The first condition involves also a correction factor D, which is subtracted to the value of the maximal correlation coefficient. The correction factor D is introduced as the Alternating Conditional Expectation (ACE) algorithm used to estimate the maximal correlation coefficient is known to work poorly when variables are nearly uncorrelated (Tibshirani, 1988). Note that no correction is used in the second condition of the test, as the estimate is assumed to be reliable when $r^*_{\mathbf{x},\mathbf{y}}$ is high. We provide an overview of this phenomenon and describe strategies to compute D in the Supplementary Material.

The second test regards the quadratic test. The nonlinear correlation is deemed significant if $p_{\text{val}} < \alpha_{\text{QTadj}}$, and as insignificant otherwise. The threshold for the test is $\alpha_{\text{QTadj}} = \alpha_{\text{QT}}/c_{\text{BQT}}$, where α_{QT} is the nominal significance level of the test and c_{BQT} is the Bonferroni correction factor. Such a correction is achieved by dividing the original significance level by the number of tests being conducted simultaneously. If m variables are available, then $c_{\text{BQT}} = m(m-1)$. The default significance level is set as in SPA (Sun, 2020a): $\alpha_{\text{QT}} = 0.01$.

The aforementioned nonlinearity significance tests are applied to all couples of variables in the dataset. However, an aggregation rule is to be chosen, as the nonlinearity property must be assigned to the whole dataset rather than to specific couples of variables. These considerations lead to propose three criteria for nonlinearity detection:

1. The "any" criterion: The dataset is deemed nonlinear if any couple of variables feature a significant nonlinear correlation (which is consistent with Sun and Braatz(2021)).
2. The "variables" criterion: The dataset is deemed nonlinear if a fraction of variables involved in a significant nonlinear relationship with at least another variable is greater than ε_{nl}.
3. The "couples" criterion: The dataset is deemed nonlinear if the fraction of couples of variables featuring significant nonlinear relationships is greater than ε_{nl}.

The latter two criteria have a significant advantage over the first criterion. Since $m(m-1)$ couples of variables are tested, and there is a nontrivial possibility of incorrectly detecting nonlinearity in the dataset due to a single false positive ("any" criterion). The probability of this occurrence increases quadratically with m. Furthermore, linear models can manage mildly nonlinear datasets by adding some more principal components/latent variables, e.g., see discussions by Dong and Mcavoy (1996), Xu et al. (1992), and Paluš and Dvořák (1992). The default value of the fraction of nonlinear variables/couples to be used in both the mentioned criteria is set as $\varepsilon_{nl} = 0.1$ as this fraction starts to exceed mildly nonlinear behavior that can still be handled by linear models.

The three criteria are compared by means of two Monte Carlo simulations, identical in settings to the simulations discussed in Section 3.2.1, with the same factors but responses being nonlinearity detection rates of the three criteria. The first study still considers the sampling distribution as one of the factors. Although it is not known *a priori* whether such distributions feature nonlinear correlation of variables or not, this study is done to test the hypothesis made in Section 3.1, namely that non-normality and nonlinearity are a tightly interconnected properties of a dataset. For the same reason, the non-normality detection rate is also included among responses. The combination of the three tests according to the rationale outlined at the end of Section 3.2.1 is used to test non-normality. Settings of all criteria are kept to default values.

A general observation emerging from both Monte Carlo studies is that the sample size is extremely important for the reliability of nonlinearity assessments. In fact, all criteria correctly deem samples from normal distribution as normal in nearly all repetitions only for $N \geq 500$. Fig. 6 shows that, as expected, the "any" criterion is the least robust, while the "couples" criterion is the most robust, being perfect in recognizing linear datasets even for $N \geq 200$. The "variables" criterion yields acceptable results for $\frac{N}{m} \geq 4$.

Considering samples drawn from other distributions (see the Supplementary Material for detailed results), all criteria are nearly perfect in detecting nonlinearity of the lognormal distribution, with the "couples" criterion sporadically exhibiting erratic behavior. Detection of the t distribution is harder, due to the mild deviation from normality. In this case, the "couples" criterion is the worst performing, while the "any" criterion is the best performing. An interesting trend can be noticed, where the "any" criterion appears to work better for low ratios of the number of observations the number of variables, making up for the lack of performance of the non-normality detection criterion in this case. However, this performance is misleading and due to lack of sufficient observations to properly characterize the data.

Results on the uniform distribution, shown in Fig. 7, are the most interesting. As expected, samples are correctly deemed non-normal, yet linear. This occurs due to the multivariate uniform distribution featuring no correlation at all. Such results show that the default thresholds for nonlinearity assessment regarding the maximal correlation coefficient, together with the default deflation approach, are adequate to not misclassify independent variables as nonlinearly correlated. The results also confirm that at least $N = 500$ is needed for the reliability of the "any" criterion, whereas the "variables" and "couples" criteria allow to lower that threshold to $N \geq 200$, though a larger number of observations is still recommended to obtain high reliability of nonlinearity detection.

Moving to the Monte Carlo study generating samples given the fraction of nonlinear variables, consider the case $f_{nl} = 0.05$. In this case, no nonlinear variables are included if $m = 10$, while only one variable is included if $m = 30$. This last occurrence yields the minimum value of the fraction of nonlinear variables, $\frac{2}{m} = 0.06667$, achieved if one single couple features nonlinear correlation. Fig. 8 highlights the importance of the ratio of the number of observations to the number of variables, as especially apparent from the results of the "variables" criterion. Finally, the "couples" criterion is the only one consistently recognizing the dataset as linear according to the set threshold.

Cases with higher f_{nl} allow to draw conclusions similar to the those already known concerning the robustness of methods. Besides the case $f_{nl} = 0.1$, where the "any" criterion appears to be a little too strict with respect to the "variable" criterion (the former has detection rates always very close to 1 even for low f_{nl}), these two criteria show similar results in all cases (see the Supplementary Material for detailed results). On the other hand, the "couples" criterion consistently misses the nonlinearity of the dataset, achieving acceptable performance only if $\frac{N}{m} \geq 200$, which is unreasonable. This lack of performance could be due to the fact that the number of couples required to overtake the threshold for this criterion varies as m^2, therefore increasing sharply with the number of variables. This makes the criterion robust to the rejection of the nonlinearity hypothesis, but overly conservative to its acceptance, therefore being prone to high false-negative rates. The case with $f_{nl} = 0.4$ is shown in Fig. 9 as an example of this behavior.

Considering all of the outcomes of the Monte Carlo studies, the "variables" criterion is chosen as the default criterion to assess nonlinearity of a dataset. The motivation is that this method shows the best tradeoff between detection rate on nonlinear datasets and the rejection rate on linear datasets, being sufficiently robust and sensitive for $N \geq 100$ and $\frac{N}{m} \geq 4$. Furthermore, this method offers a nice insight on the "intensity" of the nonlinearity of the dataset, which can be quantified by the fraction of variables involved in nonlinear relationships and by the map of variables/couples deemed nonlinear. The most prominent drawback of the selected methods is that its resolution (minimum value that the fraction of nonlinear variables can assume) degrades as the number of variables decreases.

3.2.3. Dynamics detection

One of the assumptions underlying PCA and PLS is that data do not feature any autocorrelation. Although sometimes the dynamics in data are mild enough to be represented reasonably well by a static model, the dynamics effects would remain unmodeled and would show up in the residuals.

Several functions are useful for the characterization of dynamics of a variable given a set of its observations (Box et al., 2016). The Auto-Correlation Function (ACF) characterizes the general dynamic behavior of a time series. The partial autocorrelation function characterizes the dynamics of a time series in term of optimal autoregressive models, thereby "removing" the effect of the ACF. If evaluating the interdependence of two time series is of interest, the cross-correlation function can be used. We consider the ACF to set up the dynamics detection test implemented in SPAfPM. We motivate our choice in the Supplementary Material.

In its sample versions, the ACF exploits the concept of lagged measurements and yields a coefficient for each lag order. The significance of coefficients can be evaluated using the Ljung–Box statistics (Ljung and Box, 1978), which also allows to adjust the nominal significance level by means of the Bonferroni correction (Hochberg, 1988) if more than one coefficient is tested simultaneously. In general, a variable features no significant dynamics if no coefficient is deemed significant in the ACF.

Given a time series $\mathbf{x} \in \mathbb{R}^N$ of a random process X, the (sample) ACF coefficient at lag l is defined as (Box et al., 2016)

$$r_{\mathbf{x}}(l) = \frac{c_{\mathbf{x}}(l)}{c_{\mathbf{x}}(0)} \tag{7}$$

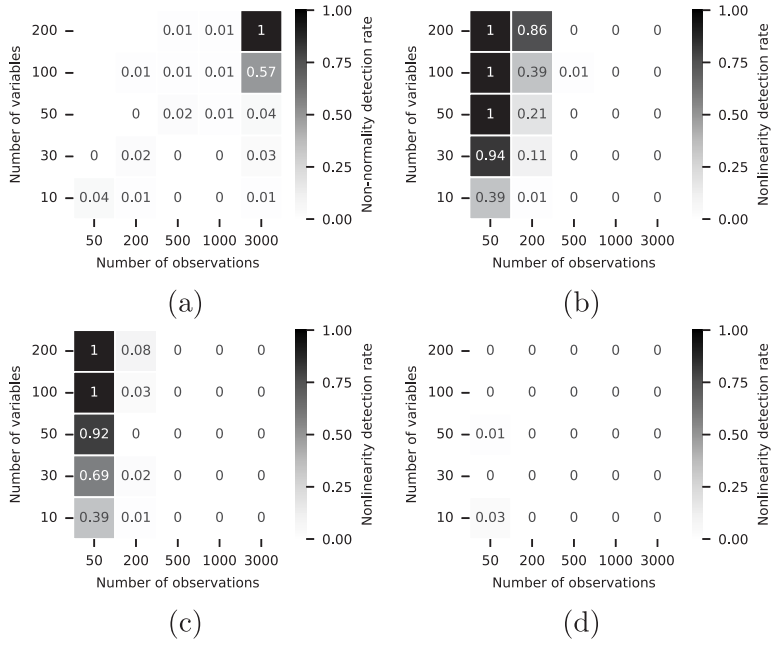

Fig. 6. Nonlinearity detection rates of the proposed criteria on samples generated from multivariate normal distributions: (a) Combination of non-normality tests, (b) "any" criterion, (c) "variables" criterion, and (d) "couples" criterion.

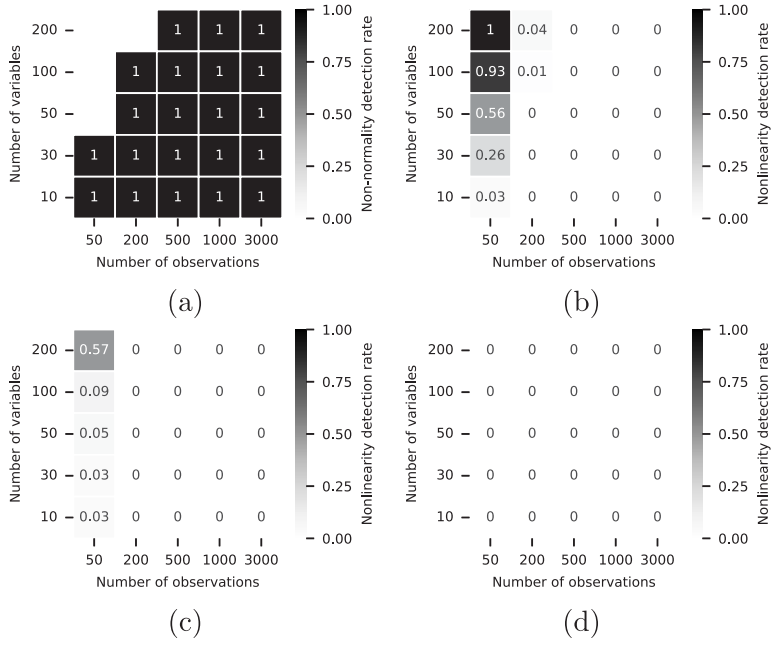

Fig. 7. Nonlinearity detection rates of the proposed criteria on samples generated from multivariate uniform distributions. The tests are in the same order as in Fig. 6.

where $c_{\mathbf{x}}(l)$ is the sample autocovariance function of the time series at lag l, defined as

$$c_{\mathbf{x}}(l) = \frac{1}{N}\sum_{n=1}^{N-l}\left[(x_n - \bar{x})(x_{n+l} - \bar{x})\right], \tag{8}$$

where $\bar{x}$ is the sample mean of the process.[2] The significance of autocorrelation coefficients can be determined using the Ljung–Box statistic, (Ljung and Box, 1978)

$$\tilde{Q}(l) = N(N+2)\sum_{k=1}^{l}\left[\frac{1}{N-k}\left(r_{\mathbf{x}}(k)\right)^2\right]. \tag{9}$$

The $\tilde{Q}(l)$ statistic is approximately distributed as a χ^2 variable with l degrees of freedom. The p-value of the statistic can be compared to the Bonferroni-adjusted significance level of the test, $\alpha_{\text{ACFadj}} = \alpha_{\text{ACF}}/c_{\text{BACF}}$, where $c_{\text{BACF}} = l$ is the number of coefficients being tested simultaneously. By default, $\alpha_{\text{ACF}} = 0.01$. The sample $\mathbf{x}$ is deemed to feature significant dynamics if at least one coefficient $r_{\mathbf{x}}(l), l \in \{1, \dots, h\}$, with $h = \min\{20, \lfloor \frac{N}{2} \rfloor - 1\}$, is deemed significant.

[2] Aside: $c_{\mathbf{x}}(0)$ is a biased version of the sample variance ($s^2_{\mathbf{x}}$) of the process.

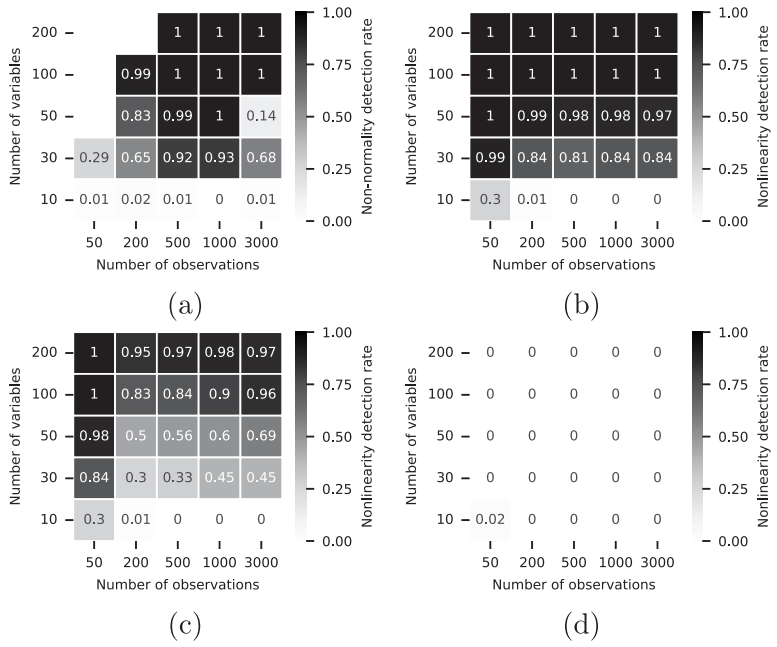

Fig. 8. Nonlinearity detection rates of the proposed criteria on samples in which 5% of the variables are nonlinear. The tests are in the same order as in Fig. 6.

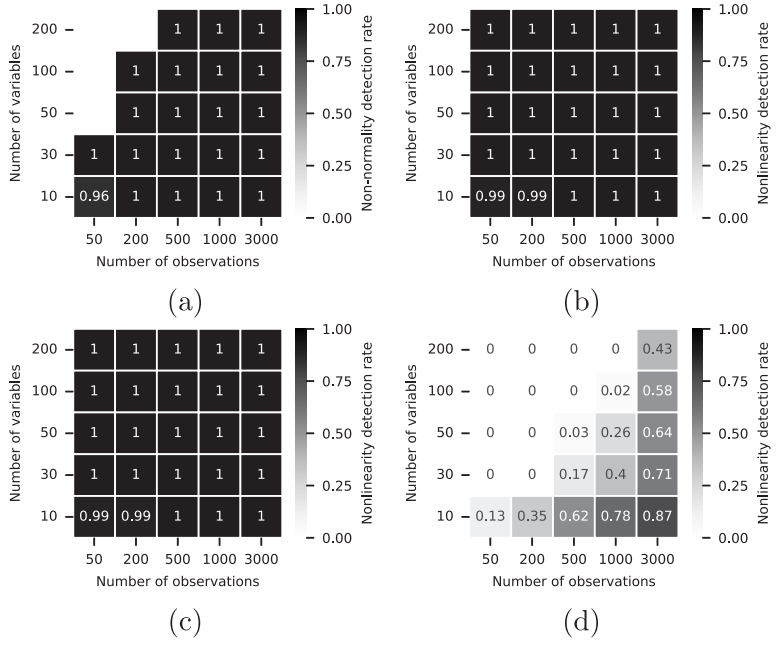

Fig. 9. Nonlinearity detection rates of the proposed criteria on samples in which 40% of the variables are nonlinear. The tests are in the same order as in Fig. 6.

As for nonlinearity detection, dynamics are tested on all variables, but is to be attributed to the whole dataset. Two criteria are proposed to detect dynamics in a dataset of N observations and m variables:

1. The "any" criterion: The dataset is deemed dynamic if any of the variables feature a significant dynamic behavior.
2. The "variable" criterion: The dataset is deemed dynamic if a fraction of variables featuring significant dynamic behavior is greater than ϵ_{dyn}.

The default value of the fraction of dynamic variables is set as $\epsilon_{\text{dyn}} = 0.1$.

In contrast to constructing regression models, where dynamics is tested on residuals of a static regression model (Sun and Braatz, 2021), dynamics in the proposed framework are assessed directly on the variables in the dataset. As argued by Ku et al. (1995), applying a static model to dynamic data can extract only static components, with the dynamics being left in the residual space. In such a case, the Q statistic is expected to carry the dynamics of the residuals, hence featuring significant autocorrelation. This point can be leveraged to propose two additional criteria for dynamics detection. First, a static model of choice is selected according to the outcomes of the nonlinearity detection criterion and the presence of dependent variables. Two versions of the model are built: Model A using the parameters corresponding to

the minimum error in cross-validation; one Model B using the one-standard-error-rule (see Section 3.3 for details). Based on this, two additional criteria for dynamics detection can be defined as:

3. The "model_min" criterion: The dataset is deemed dynamic if the Q statistic from model A features a significant dynamics.
4. The "model_oster" criterion: The same rationale of the previous criterion is adopted, but Q coming from model B.

The four proposed criteria are evaluated in a Monte Carlo study. The factors of the study are:

- The fraction of dynamic variables in the sample: $f_{\text{dyn}} \in \{0, 0.05, 0.10, 0.20, 0.40, 0.80\}$.
- The number of variables in the sample: $m \in \{10, 30, 50, 100, 200\}$.
- The number of observations in the sample: $N \in \{50, 200, 500, 1000, 3000\}$.

All combinations are tested, and 100 repetitions are performed for each combination, generating a random sample at each repetition. The sample is generated in a similar way as described in Section 3.2.1. Assume, for example, that $m = 25$ and that 30% of the variables are dynamic ($f_{\text{dyn}} = 0.3$), while the remaining 70% are static variables. The first step is to sample $m_{\text{sta}} = \lfloor 0.7m \rfloor = 17$ variables from a multivariate normal distribution with randomly generated parameters. Then $m_{\text{dyn}} = m - m_{\text{sta}} = 8$ dynamic variables are to be generated. Matrices of a random state-space model are generated using an algorithm inspired by the `drss` algorithm provided by the Systems Identification Toolbox, version 9.16, of MATLAB R2022a. The state-space model has m_{sta} inputs and m_{dyn} outputs, with the number of states randomly selected as an integer between 1 and 10 (inclusive). The state-space model generation is tuned in a way that guarantees stability of the system and that the feed-through matrix is null (no direct effect of current inputs on current outputs). The observations of the m_{sta} variables are used as inputs to run the state-space model, while the corresponding outputs are collected as the m_{dyn} dynamic variables. In order to simulate stationary processes (assumption of all the aforementioned significance assessment approaches), the state is randomly initialized and 200 more observations are sampled from the same distribution used to generate the m_{sta} variables. Such observations are used to "burn-in" the state-space model with the randomly initialized state to obtain a stationary initial state, which is then used to generate the actual N observations of dynamic variables. The 200 burn-in observations are discarded. White noise is added to each of the m_{dyn} dynamic variables sampling normal distributions with zero mean and variance such that the signal-to-noise ratio of the generated variables is 1:0.1. Finally, the m_{sta} static variables and the m_{dyn} dynamic variables are jointed to produce the sample.

Concerning the "any" and "variables" criteria, the false-positive rate of the former is higher in the cases $f_{\text{dyn}} = 0$ (see Fig. 10) and $f_{\text{dyn}} = 0.05$, while the latter consistently deems the samples as static with false-positive rate very close to the nominal significance level set for the ACF. Both the model-based criteria show good performance too.

The case $f_{\text{dyn}} = 0.10$ shows a divergence in performance of the "any" and "variables" criteria, as can be seen in Fig. 11: While the "any" criterion mostly deems samples as dynamic, the "variables" criterion prefers static models, showing erratic dynamics detection rates, which increase with increasing numbers of observations and variables. This behavior is expected as $f_{\text{dyn}} = 0.10$ is the threshold set for ϵ_{dyn}. The two model-based criteria show again similar performance to the "variables" criterion, yet yielding slightly more erratic results (no clear effect of the number of variables or of the sample size).

The "any" and "variables" criteria show similar performance in the remaining cases, with the latter being slightly more prone to deem the dataset as static than the former criterion for mild dynamics. On the other hand, both the model-based criteria show very high false-negative rates and quite erratic results. For example, Fig. 12 reports the case $f_{\text{dyn}} = 0.4$ (see the Supplementary Material for detailed results). These cases also show that long time series are required to properly characterize the dynamics in the data. A general indication could be $N \geq 500$.

Based on these outcomes, the "variable" criterion is selected as the default dynamics assessment method of SPAfPM. As for the analogous criterion for nonlinearity, this criterion achieves the best tradeoff between robustness and sensitivity, while also offering a nice interpretation. The criterion is subject to the same drawback nonetheless, namely, poorer resolution as the number of variables decreases.

3.3. Cross-validation procedure

The data analytics triangle in Fig. 2 allows the user to determine the most suitable model, or subset of models, based on the characteristics of the data at hand. In case a subset of models is suggested, there needs to be a procedure to determine which is the best according to performance on NOC data. Additionally, the optimal hyperparameters for each of the models need to be chosen to provide a fair comparison between them. A commonly used method for other smart data analytics approaches for this situation is to use cross-validation (Stone, 1974; Allen, 1974).

The use of cross-validation is well established in some cases. Considering, for instance, the case of regression, prediction performances of various models on a validation dataset (not used for model fitting) can be evaluated using the mean squared error as performance index (Sun and Braatz, 2021). Similarly, the accuracy can be evaluated on validation datasets for the case of supervised classification (Mohr et al., 2022). However, in the case of fault detection, it is not trivial to define a good figure of merit to quantify performance of a model (Camacho and Ferrer, 2014).

This problem can be tackled bearing in mind that the aim of cross-validation is to optimize the generalization performance of the model, therefore the performance index that is used should be consistent with the modeling objective (Camacho and Ferrer, 2014). Typically, the fault detection algorithms are evaluated on how often they incorrectly qualify NOC observations as faults (Type I error), and how often they miss faulty observations (Type II error). For the Type II error, it is necessary to have data from faulty operating conditions. While it is not difficult to produce such data using simulators, it is uncommon that comprehensive datasets including all possible faults are available in real, industrial applications (even though this might be the case for some specific process). Therefore, SPAfPM relies on the restrictive assumption that only NOC data are available for model calibration and selection.

However, the Type I error can be used as an evaluation metric for the validation datasets, which is a "good practice" frequently mentioned in the fault detection literature: the validation Type I error rate (i.e., that fraction of normal observations detected as faulty on a validation/testing NOC dataset) should be as close as possible to the nominal significance level of control limits, α. This point is explicitly suggested by a number of studies (Camacho et al., 2016; Camacho and Picó, 2006b; Ramaker et al., 2006; Yoon and MacGregor, 2004). For example, Ramaker et al. (2006) state that "it is useful to check whether the fraction of out-of-control signals for a given data set is close to α in case the control charts are set at this significance level. ... The performance of a chart in terms of Type I error is good if α observed is close to α". As another example, Yoon and MacGregor (2004) suggest that "By calculating the false alarm rate during normal operating conditions for the testing set and comparing it against the level of significance upon which the threshold is based, one can measure the robustness of a fault detection method". This condition is also regarded as essential if fault detection performances of multiple models are to be compared (Reis et al., 2021; Rato and Reis, 2013; Camacho et al., 2009). While all of the aforementioned studies suggest that matching the Type I error to the nominal α by manually adjusting control limits, however, the studies do not offer any guideline on how to select the model's hyperparameters, such as the number of principal components,

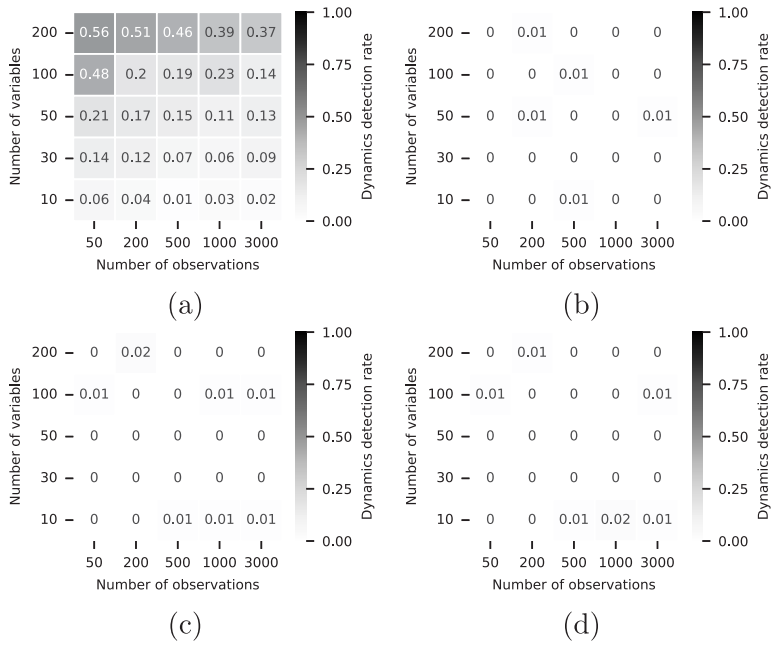

Fig. 10. Dynamics detection rates of the proposed criteria on samples in which 0% of the variables are dynamic: (a) "any" criterion, (b) "variables" criterion, (c) "model_min" criterion, and (d) "model_oster" criterion.

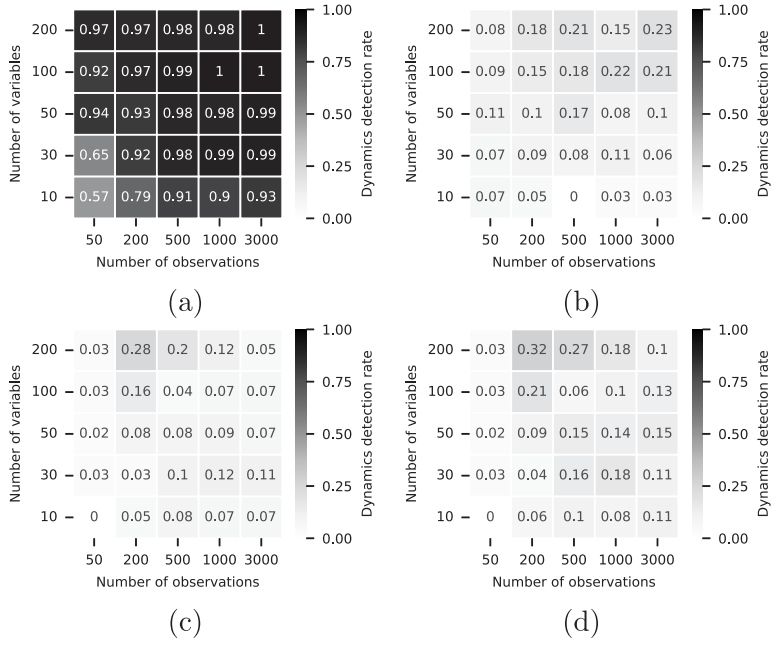

Fig. 11. Dynamics detection rates of the proposed criteria on samples in which 10% of the variables are dynamic. The tests are in the same order as in Fig. 10.

consistently to the modeling objective, in the sense of Camacho and Ferrer (2014).

The Type I error can be used as an evaluation metric for the validation dataset nonetheless. Ideally, the Type I error should be as close to α as possible, which is referred to as *rigorous approach* in the model-aided adulteration detection literature (Rodionova et al., 2016) in which the objective is basically the same as of fault detection in industrial systems. A similar approach is also used by KDE-CVA or by SVDD, with (S.36) showing that the control limit is chosen based on the significance level α (Odiowei and Cao, 2010). Ultimately, our objective is to choose the algorithm that yields a Type I error approximately as often as α. Performing model selection on the basis of this index is consistent with the monitoring objective, as suggested by Camacho and Ferrer (2014), and automates the fulfillment of the "criterion for monitoring performance" suggested by Camacho et al. (2016), Ramaker et al. (2006), and Yoon and MacGregor (2004). In this way, an empirical, possibly inconsistent model selection, followed by an empirical adjustment of control limits, is automated in a single, consistent operation.

However, it is worth mentioning that the use of faulty data for model selection in process monitoring, the so-called compliant approach (Rodionova et al., 2016), may be preferred in some cases. Model featuring a high degree of complexity may particularly benefit from such an approach. We carry out a brief discussion of rigorous vs. compliant cross-validation in the Supplementary Material, which is relevant for the case studies reported in Section 4.3 and Section 4.4.

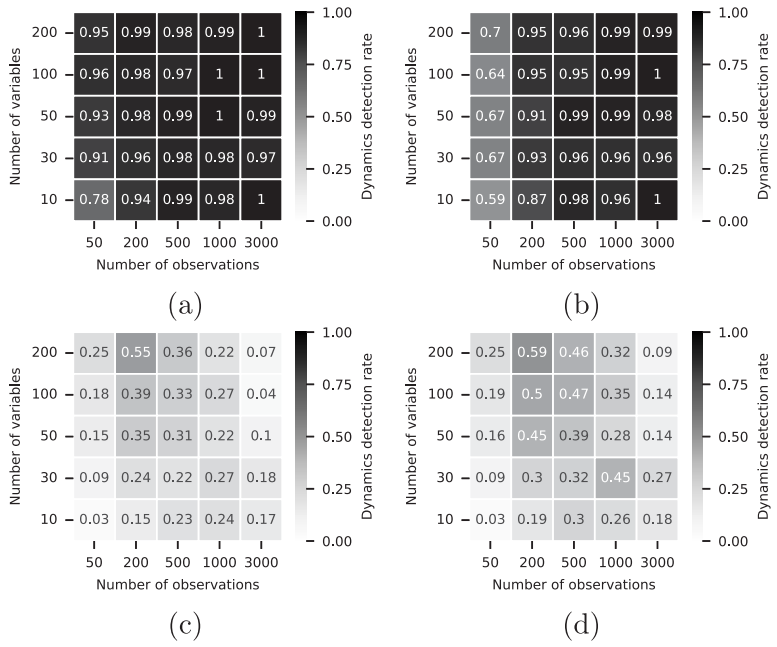

Fig. 12. Dynamics detection rates of the proposed criteria on samples in which 40% of the variables are dynamic. The tests are in the same order as in Fig. 10.

Considering, for instance, PCA, a Type I error occurs if either the T^2 or the Q statistic crosses the threshold for normal operating conditions, which is equivalent to minimizing the objective function

$$J_\alpha^{\text{PCA}} = \left| \frac{1}{N_{\text{val}}} \sum_{n=1}^{N_{\text{val}}} \left[g_\alpha^{\text{PCA}}(\mathbf{x}_n) \right] - \alpha \right| \tag{10}$$

in cross-validation, where N_{val} is the number of observations in the validation dataset, and g_α^{PCA} is the fault indicator function for PCA, defined as

$$g_\alpha^{\text{PCA}}(\mathbf{x}) = \begin{cases} 0, & \text{if } T^2(\mathbf{x}) \le T^2{}_{\text{lim}}|_\alpha \textbf{ and } Q(\mathbf{x}) \le Q_{\text{lim}}|_\alpha \\ 1, & \text{if } T^2(\mathbf{x}) > T^2{}_{\text{lim}}|_\alpha \textbf{ or } Q(\mathbf{x}) > Q_{\text{lim}}|_\alpha \end{cases} \tag{11}$$

Most of the models included in SPAfPM share this indicator function, being based on the same fault detection statistics. Only differences are with SVDD and CVA-based approaches, where the statistics are different. The rationale of the fault indicator function is the same nonetheless.

In order to achieve a more robust model, the one-standard-error rule (Hastie et al., 2009; Filzmoser et al., 2009) is applied. Instead of just choosing the set of hyperparameters that yields the minimum value for the expression (10), the set of hyperparameters yielding the most parsimonious model whose cross-validation metric is still within one standard error from the minimum is chosen. Usually, this approach yields a conservative estimate and leads to the selection of models that are more robust and less prone to overfitting (e.g., as discussed by Sun and Braatz (2021)).

For most models, a repeated k-fold cross-validation (Burman, 1989) procedure can be used. In this case, the data are randomly split into k folds containing the roughly N/k observations each. Subsequently, $k-1$ folds are used to train the model, while the remaining fold is used as validation data. This is repeated for each one of the folds. Repeated k-fold cross validation means that this procedure is repeated several times for different k-fold splits of the dataset (Sun and Braatz, 2021).

This approach cannot be applied to dynamic models because the random splitting results in the loss of dynamic effects (Bergmeir and Benítez, 2012). Instead, the so-called growing-window cross-validation (Makridakis, 1990) is employed for dynamic models. Data are first split into k blocks of contiguous observations, with no alteration in their order. At the first iterations, the first block is used to build a model, and the second block is used as the validation dataset. For the second iteration, the first two blocks are used to train a model, and the third block is used for validation and so forth. Overall this results in different training sizes, entailing more variability in the cross-validation results.

4. Case studies

This section illustrates the effectiveness of the proposed approach for a number of case studies. The first is simple linear, static dataset designed for illustration purposes. The second is the TEP, which a widely used benchmark in the process monitoring literature. A realistic simulation of a complex process, the continuous filtration and drying of paracetamol, is the third case study. Industrial data from a metal etching process are used for the fourth case study. All case studies consider both with and without quality variables. Repeated k-fold cross validation with random splitting of observation is used to identify hyperparameters of the static models.

4.1. Simulated linear static data

This simple numerical example illustrates the proposed framework in a controlled environment. NOC data are generated by sampling from a multivariate normal distribution with $m = 15$ variables (featuring a defined correlation structure). $N = 600$ observations are sampled and used as NOC data. An additional variable is included in the dataset, as quality variable for testing the performance of quality-relevant monitoring. The quality variable is computed as a linear combination of the m variables in the "process" dataset. Gaussian noise with zero mean and variance selected in such a way that the signal-to-noise ratio is $1:0.05$ is added to the quality variable.

Three faulty datasets used for testing are

Fault 1 A step change in the mean vector of the distribution.

Fault 2 A change in the correlation structure of the input variables.

Fault 3 The onset of dynamic behavior (i.e., autocorrelation among observations).

For each faulty dataset, 200 observations are samples from the NOC distribution and $N_{\text{F}} = 1000$ faulty observations are generated afterwards.

Table 2
Linear case study: Type I and II error rates for Faults 1, 2, and 3 for all fault detection methods that do not consider dependent variables separately.

Fault #		PCA	DPCA	KPCA	DKPCA	SVDD
1	Type I error rate	0.040	0.020	0.000	0.000	0.005
	Type II error rate	0.000	0.913	1.000	1.000	0.983
2	Type I error rate	0.020	0.000	0.000	0.015	0.010
	Type II error rate	0.042	0.999	1.000	1.000	0.999
3	Type I error rate	0.020	0.025	0.000	0.010	0.005
	Type II error rate	0.130	0.950	1.000	1.000	0.996

Therefore, each fault kicks in after 200 NOC observations. See the Supplementary Material for additional details on the data generation.

The calibration dataset for the proposed smart data analytics framework consists only of the NOC data and does not include any faulty data. The NOC data is first analyzed to determine the relevant data characteristics, which are used to pre-select suitable fault detection methods. Then, the candidate models are evaluated by cross-validation to tune their hyperparameters and to select the best performing model. The dataset containing the faults are treated as testing data as to evaluate the rates of both Type I error (NOC observation incorrectly deemed faulty) and Type II error (faulty observations incorrectly deemed NOC). A fault is detected whenever either of the model statistics crosses the relevant control limit, coherently with the cross-validation procedure elucidated in Section 3.3. Both the cases with and without quality variables are considered.

The criteria introduced in Section 3.2 are used to characterize the NOC dataset used for model building (note that the NOC dataset is the same regardless of the presence of dependent variables in this case study). The results are:

- Royston's test is selected to assess non-normality. The dataset is deemed normal with a p-value of 0.7537. The dataset is deemed normal also by all the non-selected tests.
- According to the "variables" criterion, the dataset is deemed linear with a fraction of variables involved in nonlinear relationships equal to 0. All approaches to deflate the maximal correlation coefficient yield the same result. The dataset is deemed linear even if no deflation is used.
- According to the "variables" criterion, the dataset is deemed static with a fraction of dynamic variables equal to 0.

A linear and static method is appropriate to model the NOC data. We consider cases with and without dependent variables. When no dependent variable is considered separately, the proposed framework selects the PCA algorithm according to Fig. 2. Cross-validation with 5 folds and 10 repeats is used to determine the hyperparameters, yielding $a = 3$. In the case where dependent variables are considered separately, PLS is recommended as the most suitable algorithm. The cross-validation procedure concludes that $a = 3$ should be used.

PCA achieves a Type I error of 0.040 and a Type II error of 0.000 for the Fault 1. An overview of the different algorithms not considering dependent variables separeately is shown in Table 2. The T^2 and the Q statistic for PCA on Fault 1 are shown in Fig. 13. The performance of the recommended algorithm is overall very good. PCA has strong performance in terms of Type I error in validation and has by far the lowest Type II error.

For the case where dependent variables are considered separately in the fault detection procedure, PLS results in a Type I error of 0.035 and a Type II error of 0.000 for Fault 1 (Table 3). The T^2 and Q statistics for PLS on Fault 1 are shown in Fig. 14. The performance of the recommended algorithm is overall very good, with low Type I and II errors. CVA and KDE-CVA show similar performance to PLS, whereas DPLS, KPLS, and DKPLS have large Type II error rates.

4.2. Tennessee eastman process

The TEP is a well-known benchmark for process monitoring applications. Many different algorithms have been applied to the TEP to evaluate their performance in fault detection scenarios (Tien et al., 2004; Yin et al., 2011; Russell et al., 2000; Cui et al., 2008; Wang and Shi, 2014; Jia and Zhang, 2016; Tien et al., 2012). The simulator was developed by the Eastman Chemical Company to represent a real industrial chemical process consisting of a reactor, condenser, compressor, separator, and stripper (Downs and Vogel, 1993). The dataset generated by Chiang et al. (2001) is used in this study. NOC data, including $N = 500$ observations of the 52 process variables, were obtained by Chiang et al. (2001) running the simulator in normal operating conditions (without any faults). 21 additional simulations were run with a particular type of fault (pre-implemented in the simulator) to obtain 21 faulty datasets for testing.[3] All the faulty datasets consist of 160 observations of NOC data and $N_F = 800$ additional observations of faulty operating conditions, which can be the result of different changes, such as different temperatures or varying feed ratios. The detection difficulty of the different faults varies significantly and it is known that certain algorithms work well on some faults, but not on others (Russell et al., 2000). Additionally, Faults 3, 9, and 15 are known to be undetectable (Chiang et al., 2001) and are therefore not considered in the analysis carried out here. As for the previous case study, the faulty operating data are used in testing to compute both the Type I and II error rates.

We consider both possible cases in terms of dependent variables. The TEP features 52 variables consisting of 11 manipulated variables and 41 process measurements (Downs and Vogel, 1993). When algorithms not considering dependent variables separately (such as PCA) are applied to the TEP, then all the variables are typically included in the dataset (Chiang et al., 2001). When dependent variables are accounted for, different publications consider different variables to be either input or dependent variables. However, most publications agree to consider the 11 manipulated variables and the first 21 process measurements as inputs (Oliveri et al., 2014; Jiao et al., 2015; Jia and Zhang, 2016). There are different options for the dependent variables: this case study considers the mole percent of component G in Stream 9 as the dependent variable, as in Oliveri et al. (2014), Jiao et al. (2015).

The criteria introduced in Section 3.2 are used to characterize the NOC dataset used for model building.[4] The results are:

- Royston's test is selected to assess non-normality. The dataset is deemed non-normal with a p-value basically equal to 0. This result is due to the marginal distributions of some variables. For

[3] This study uses the same faults as defined by Downs and Vogel (1993). It can be argued that not all of the deviations from NOC are necessarily faults in equipment. While that argument has merit, the TEP is still used here as it has been the most widely used dataset for comparing fault detection methods, and does provide a wide range of extent of operational deviation from NOC. As most commonly used in the literature, the term "fault" is used in this article to indicate any deviation from NOC, irrespective of whether it would be an active concern in a real chemical plant.

[4] The NOC dataset differs in the cases with and without dependent variables in this case study.

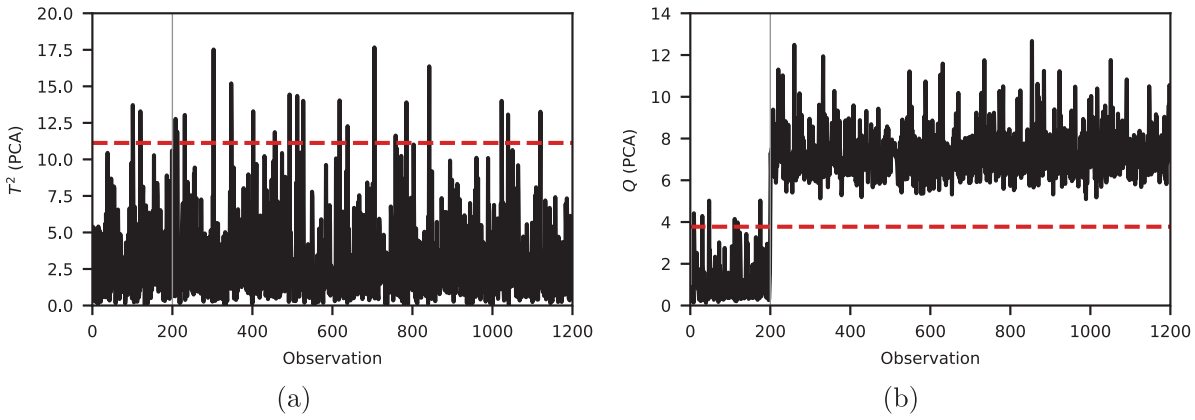

Fig. 13. Linear case study: (a) T^2 statistic and (b) Q statistic for PCA applied to Fault 1. The dashed red line is the χ^2 control limit. The fault occurs at observation 200 (vertical line) which is detected by the Q statistic.

Table 3
Linear case study: Type I and II error rates for Faults 1, 2, and 3 for all methods that consider dependent variables separately.

Fault #		PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
1	Type I error rate	0.035	0.025	0.010	0.005	0.035	0.030
	Type II error rate	0.000	0.930	0.999	1.000	0.000	0.000
2	Type I error rate	0.030	0.015	0.005	0.000	0.010	0.010
	Type II error rate	0.047	0.995	1.000	1.000	0.000	0.000
3	Type I error rate	0.030	0.020	0.000	0.005	0.010	0.010
	Type II error rate	0.121	0.956	1.000	1.000	0.003	0.003

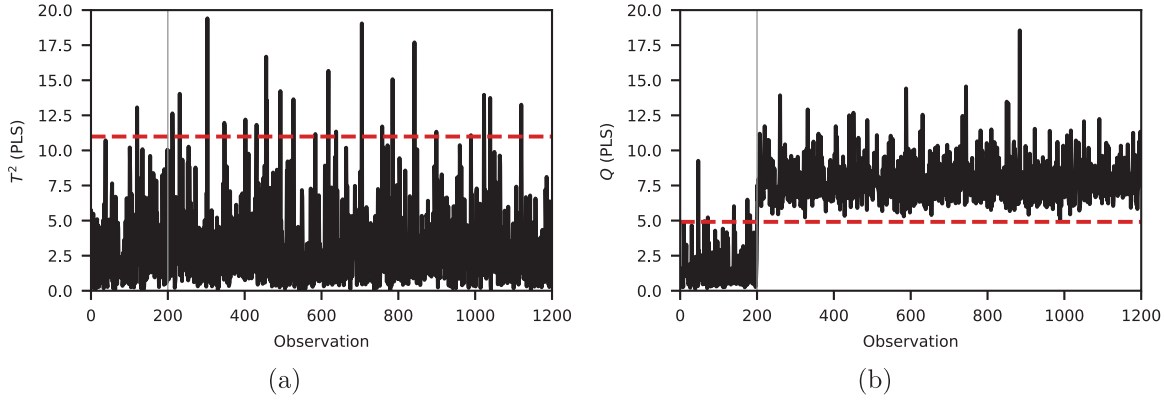

Fig. 14. Linear case study: (a) T^2 statistic and (b) Q statistic for PLS applied to Fault 1. The dashed red line is the χ^2 control limit. The fault occurs at observation 200 (vertical line) which is detected by the Q statistic.

instance, variables 37 to 41 show "stair" profiles as a result of their lower sampling frequency.

- According to the "variables" criterion, the dataset is deemed linear with a fraction of variables involved in nonlinear relationships equal to 0 for both cases with and without dependent variables. Such a result is in accordance with the literature (Sun, 2020b). This result, combined with the non-normality detection criterion, also allows to conjecture that variables are either uncorrelated or only linearly correlated; an inspection of the maximal correlation and linear correlation matrices reveals that variables are mostly uncorrelated, with few cases of linear correlation (due to linear constraints among variables imposed by material balances of the process). All approaches to deflate the maximal correlation coefficient yield the same results in this case.
- According to the "variables" criterion, the dataset is deemed dynamic with a fraction of dynamic variables equal to 0.788 (41 dynamic variables out of 52) in the case without dependent variables, and equal to 0.667 (22 dynamic variables out of 33) when output variables are considered. This result is also in accordance with literature (Sun, 2020b).

The preliminary data interrogation indicates that there are always dynamics and no nonlinearity present in the NOC data. Considering the case without separate dependent variables, the data analytics triangle of SPAfPM (Fig. 2) suggests DPCA as the best algorithm. A 5-fold timeseries cross-validation is used to determine the hyperparameters of DPCA, yielding one principal component and one lag: $a = 1$ and $h = 1$. In the case with separate dependent variables, CVA and DPLS are recommended. SPAfPM performs a 5-fold timeseries cross-validation procedure to designate the best model, as described in Section 3.3. Fig. 15 shows the distribution of the Type I error rate in validation for CVA and DPLS. The validation errors are lower for CVA on average and lower errors are more frequent. SPAfPM selects CVA as final model, with $a = 1$ and $h = j = 1$ as hyperparameters.

4.2.1. Fault 1

Fault 1 is one of the most frequently analyzed faults. In this case, the ratio of Components A and C in Stream 4 is varied in the form of a step change, with Component C increasing and Component A decreasing (Chiang et al., 2001).

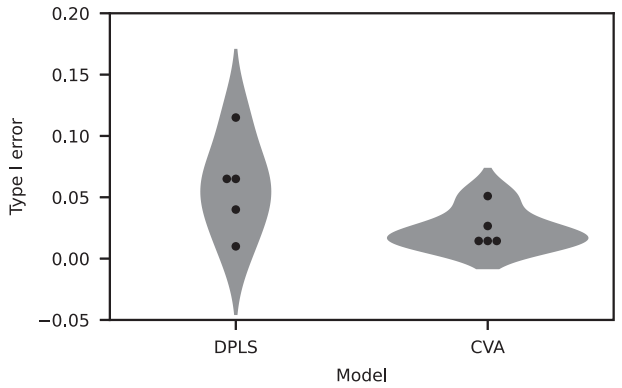

Fig. 15. TEP case study: Violin plot of the validation errors for DPLS and CVA. Each black dot marks the validation error for one of the cross-validation folds.

Table 4
TEP case study: Type I and II error rates for Fault 1 for all methods not considering dependent variables separately.

	PCA	DPCA	KPCA	DKPCA	SVDD
Type I error rate	0.019	0.006	0.006	0.000	0.025
Type II error rate	0.004	0.001	1.000	0.098	0.003

The Type I and II errors for all methods without separate dependent variables is shown in Table 4. The suggested method, DPCA, is the second best performing algorithm with a Type I error rate of 0.006 and is the best performing algorithm in terms of Type II error with a rate of 0.001. With regard to the T^2 and Q statistics (Fig. 16), the T^2 statistic partly crosses the threshold after the fault occurs, but is not consistent. The Q statistic continuously detects the fault and shows good performance for the normal operating conditions in the beginning as well.

The Type I and II errors for methods that include the dependent variables separately in the data are shown in Table 5. CVA has a relatively high Type I error rate, but detects all of the faulty operating data resulting in a Type II error rate of 0. DPLS (i.e., the alternative method suggested by SPAfPMs) performs well in terms of both Type I and II error rates. The T^2 and the T_r^2 both detect Fault 1 perfectly (Fig. 17). However, the T_r^2 statistic is fairly sensitive leading to a higher Type I error rate.

4.2.2. Fault 5

Fault 5 is also a commonly used to assess fault detection methods. In this case, the condenser cooling water inlet temperature experiences a step change causing a change of the reactor cooling water flow rate (Chiang et al., 2001). The fault rates for methods not considering dependent variables separately are shown in Table 6. DPCA has a higher Type I error, but is the second best option for the Type II error. However, the Type II error is still high. The reason for this result is that methods that do not consider dependent variables are not suitable for detecting Fault 5 (Chiang et al., 2001).

If dependent variables are considered separately, our algorithm recommends CVA. CVA yields a Type I error rate of 0.044, comparing well to other methods (Table 7). In terms of Type II error, CVA and KDE-CVA clearly outperform the other methods and achieve a flawless Type II error rate of 0. The T^2, T_r^2, and Q statistic for CVA are shown in Fig. 18. In this case, only the T_r^2 statistic is capable of consistently detecting the fault after its occurrence, which is why only CVA and KDE-CVA show such a good performance on this fault. This case shows that the recommendation of CVA based on the characteristics is appropriate.

4.2.3. Overall performance

For data characterized as being dynamic and linear, the proposed algorithm consistently suggests DPCA for models not considering dependent variables separately, and CVA for models considering dependent variables separately. Obviously, the faults vary and there is not one algorithm that handles all of the faults perfectly (Russell et al., 2000). However, we are interested in evaluating how the proposed algorithms compare to the other algorithms for the detectable faults in terms of the Type I and II errors. In terms of the Type I error, all algorithms showed similar performance, with kernel-based methods tending to perform slightly better on average. The reason for this result is that kernel-based methods are very good at fitting a model; unfortunately, kernel-based methods are also prone to overfitting. This behavior helps achieve very low Type I error rates, but results in high Type II error rates. Table 8 shows the Type II error rates for all algorithms not considering dependent variables, in which DPCA is observed to be the best performing algorithm in 12 out of 18 cases, with performance nearly identical to the best methods in the remaining 6 cases. Similarly, Table 9 shows the Type II error rates for all algorithms considering dependent variables. In this case, CVA is the best performing algorithm in 16 out of 18 cases. These results demonstrate the effectiveness and strong performance of the proposed smart data analytics approach for selecting the best method for process monitoring for this case study.

4.3. Continuous filtration and drying of paracetamol

Destro et al. (2021) developed a detailed, highly nonlinear, mechanistic model of a continuous filtration and drying process for an Active Pharmaceutical Ingredient (API). ContCarSim (Destro et al., 2022), a simulator implementing such model, is freely available on GitHub (Destro, 2022). The modeled process is carried out in a revolving carousel unit with five ports. A slurry containing API crystals, the mother liquor, and the solvent is loaded in port 1; vacuum-driven deliquoring takes place in ports 2 and 3; port 4 is used for drying the crystals under a flow of hot air; the dry crystal cake is discharged in port 5. Fouling of the filter meshes is simulated as well and an automatic cleaning routine is implemented by the simulator. Measurements from $m = 8$ sensors installed on the actual machine used for model development are returned by the simulator. The reader is referred to the original publications for details on the model (Destro et al., 2021) and on the simulator (Destro et al., 2022).

ContCarSim has been used to generate the $N = 1260$ observations in the NOC datast, which gather all the $m = 8$ process variables. To test also the performance of quality-relevant monitoring, one of the simulation states – the solvent concentration in the cake being processed – is selected as a quality variable to characterize the product. Gaussian noise is added to such a state to simulate noise of a real measurement.

ContCarSim comes with two pre-implemented faults, called "disturbance scenarios" in the simulator. Scenario number 1, a ramp change in the feed slurry concentration, is selected to generate the faulty dataset. The simulator is implemented in such a way that the fault onsets after a given simulation time. Therefore, the first 210 observations of the faulty dataset are NOC, after which $N_F = 1410$ faulty observations follow. See the Supplementary Material for additional details on the data generation.

The criteria in Section 3.2 are used to characterize the NOC dataset used for model building.[5] The results are:

- Royston's test is selected to assess non-normality. The dataset is deemed non-normal with a p-value essentially equal to 0. The dataset is deemed non-normal also by all the non-selected tests.

[5] The NOC dataset is the same regardless of the presence of dependent variables in this case study.

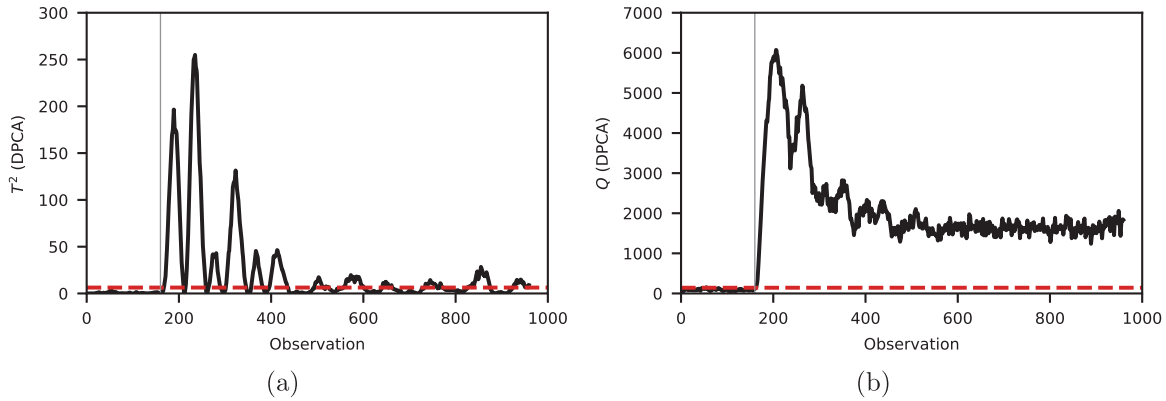

Fig. 16. TEP case study: (a) T^2 statistic and (b) Q statistic for DPCA applied to Fault 1. The dashed red line represents the χ^2 control limit. The fault occurs at observation 160 (vertical line). The Fault is quickly detected by the Q statistic.

Table 5
TEP case study: Type I and II error rates for Fault 1 for all methods considering dependent variables separately.

	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
Type I error rate	0.013	0.031	0.000	0.019	0.050	0.000
Type II error rate	0.001	0.000	0.008	0.006	0.000	0.003

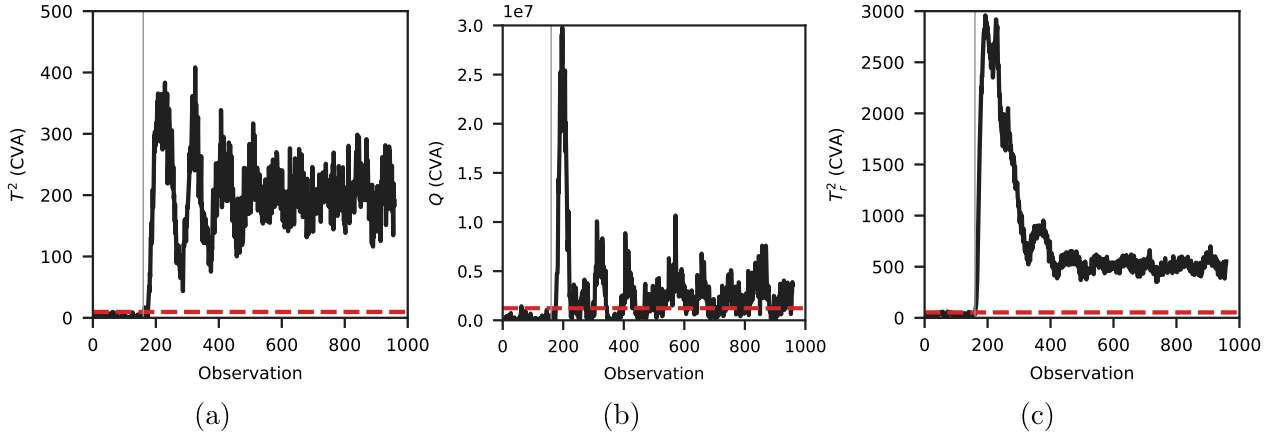

Fig. 17. TEP case study: (a) T^2 statistic, (b) Q statistic, and (c) T_r^2 statistic for CVA applied to Fault 1. The dashed red line is the χ^2 control limit. The fault occurs at observation 160 (vertical line).

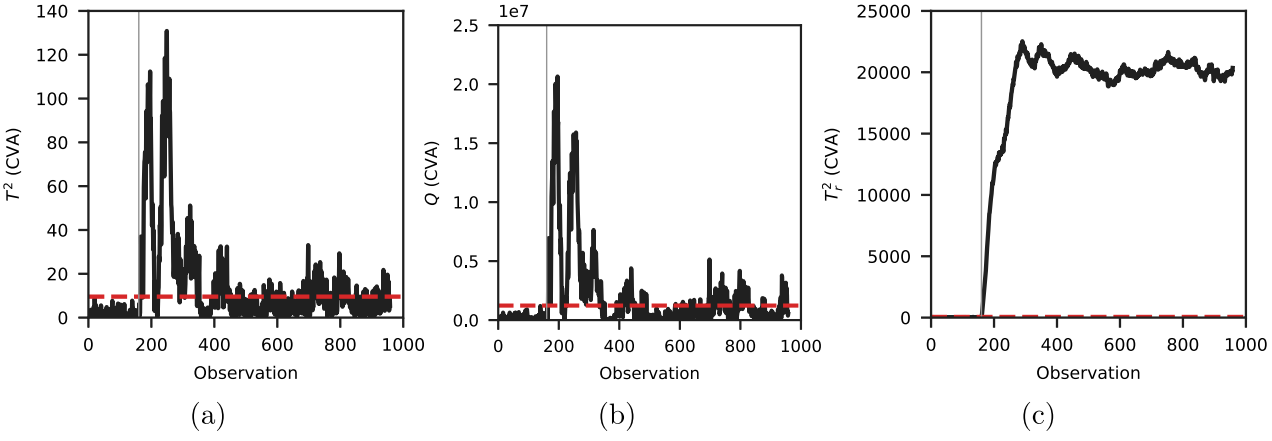

Fig. 18. TEP case study: (a) T^2 statistic, (b) Q statistic, and (c) T_r^2 statistic for CVA applied to Fault 5. The dashed red line is the χ^2 control limit. The fault occurs at observation 160 (vertical line). The T_r^2 statistic consistently detects the fault.

Table 6
TEP case study: Type I and II error rates for Fault 5 for all methods not considering dependent variables separately.

	PCA	DPCA	KPCA	DKPCA	SVDD
Type I error rate	0.038	0.056	0.019	0.032	0.038
Type II error rate	0.606	0.578	0.927	0.630	0.551

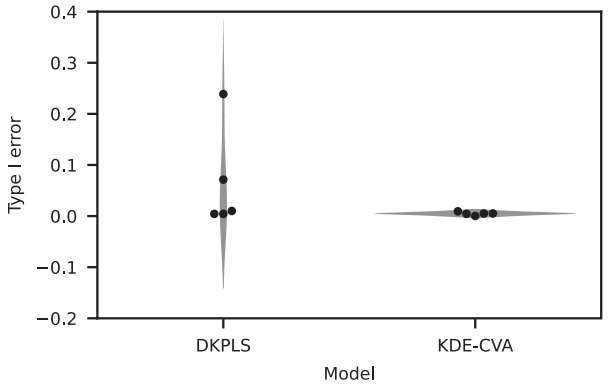

Fig. 19. ContCarSim case study: Violin plot of the validation errors for KDE-CVA and DKPLS. Each black dot marks the validation error for one of the folds.

- According to the "variables" criterion, the dataset is deemed nonlinear with a fraction of variables involved in nonlinear relationships equal to 0.625 (5 variables out of 8). This result is expected due to the high nonlinearity of the process model, and to the absence of a control system (the simulator ran in "open-loop mode"). All approaches to deflate the maximal correlation coefficient yield the same result.
- According to the "variables" criterion, the dataset is deemed dynamic with a fraction of dynamic variables equal to 0.625 (5 dynamic variables out of 8).

The above results indicate that nonlinearity and dynamics need to be accounted for when building the model of the NOC data. Similar to the previous case study, both cases without and with separate dependent variables are considered. In the former scenario, SPAfPM selects DKPCA according to Fig. 2. Cross-validation for dynamic data with 5 folds is used to determine the hyperparameters, which were $a = 1$, $h = 1$, and the radial basis function kernel with $\sigma = 50$. In the case that dependent variables are considered separately, DKPLS and KDE-CVA are selected by SPAfPM as the most suitable algorithms. Cross-validation for dynamic data with 5 folds is used for model selection and hyperparameter tuning. Based on the error distributions in Fig. 19, KDE-CVA performs better both in terms of mean validation error and in terms of variance of the validation errors for the different folds. Consequently, KDE-CVA is selected as the most suitable algorithm. The hyperparameter tuning for KDE-CVA yields $a = 3$, $h = j = 3$, and kernel bandwidth scaling factors $\xi_{T^2} = \xi_{T_r^2} = 3$, and $\xi_Q = 5$. However, DKPLS is considered as a possible alternative model (see the discussion in the Supplementary Material); hyperparameters for DKPLS result in $a = 1$, $h = 1$, and the radial basis function kernel with $\sigma = 50$.

DKPCA for the case without dependent variables results in a Type I error of 0.100 and a Type II error of 0.124. An overview of the different algorithms not considering dependent variables separately is shown in Table 10. The T^2 and the Q statistics for DKPCA are shown in Fig. 20. The performance of the selected algorithm is overall very good. DKPCA is the second best performing algorithm for the Type II error (being fundamentally equivalent to the best model, DPCA) and performs well for the Type I error. Only SVDD performs significantly better for the Type I error.

In the case that dependent variables are considered separately, the overall performance of the algorithm selected by SPAfPM, KDE-CVA, is compared to the other algorithms considering dependent variables in Table 11. The different statistics for KDE-CVA are shown in Fig. 21. The performance of the recommended algorithm is overall very good: KDE-CVA yields reasonable Type II error and shows good performance for the Type I error. The alternative model, DKPLS, performs marginally better. This example further illustrates that, as discussed in the Supplementary Material, both models can be recommended in the case that all characteristics are present.

4.4. Metal etching process

This last case study is based on data collected in an industrial plasma etch process for semiconductor manufacturing (Wise et al., 1999). The dataset can be freely downloaded (Eigenvector Research, Inc., 2024). Wafers are etched in a recipe-driven batch process carried out in a commercial Lam 9600 plasma etch machine. Integrated sensors perform online measurements collected to build the dataset, which includes 19 variables (plus the timestamp of measurements and a numerical identifier of the processing phase). Among the variables, two are binary variables (discrete with two levels): "RFB reflected power" and "TCP reflected power", as named by Wise et al. (1999). Also, many variables appear to have discrete values due to limited precision of the sensors. The complete dataset collects 108 batches under normal operating conditions. Furthermore, 21 wafers are manufactured under faulty operating conditions. For a detailed description of the process and data, refer to Wise et al. (1999).

Given the richness of features, this dataset quickly became a benchmark for evaluating novel batch process monitoring methods (Goodlin et al., 2003; Camacho and Picó, 2006a; Chen and Zhang, 2010; He and Wang, 2011; Wang and Yao, 2015; Lv et al., 2018; Du, 2019; Azamfar et al., 2020). This dataset features a high degree of correlation among variables (Cherry and Qin, 2007) and well-defined multiphase dynamics (Camacho and Picó, 2006a), and the distribution of the data is highly non-normal (Chen and Zhang, 2010). Given the presence of correlated variables and the non-normality of the data, a significant percentage of variables is expected to be involved in nonlinear relationships.

For the application of SPAfPM, one single calibration batch has been selected: the first one, named `l2901.txm` in the `MACHINE_Data.mat` dataset, which contains $N = 112$ observations. Faults 1, 10, and 16 – named `TCP+50`, `TCP+30`, and `TCP−15`, respectively (Wise et al., 1999) – are selected for testing. Three faulty datasets are obtained stacking a different NOC batch (file `l2902.txm` in the dataset), counting 107 observations, and the three aforementioned faults (files `l2915.txm`, `l3120.txm`, and `l3318.txm`, respectively). The faulty datasets contain 210, 207, and 207 observations, respectively, and all of the faults onset at observation 107.

As with the other case studies, both scenarios with and without dependent variables are assessed. In the former case, all $m = 19$ variables are considered. In the latter, variable 10 – the "phase error" in Wise et al. (1999) – is selected as the quality variable, while the remaining 18 are kept as process variables. The two aforementioned discrete variables are actually considered as such in the dataset assessment phase. On the other hand, variables that can take more than two discrete levels are considered numerical, as this occurrence is due to limited sensor precision.

Criteria introduced in Section 3.2 are used to characterize the NOC dataset used for model building.[6] The results are:

- Royston's test is selected to assess non-normality. The dataset is deemed non-normal with a p-value basically equal to 0. This result was expected due the presence of binary variables, and to the fact that many variables vary on discrete levels due to limited measurement accuracy.

[6] The NOC dataset differs in the cases with and without dependent variables in this case study.

Table 7
TEP case study: Type I and II error rates for Fault 5 for all methods considering dependent variables separately.

	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
Type I error rate	0.038	0.069	0.000	0.025	0.044	0.000
Type II error rate	0.616	0.565	0.702	0.639	0.000	0.000

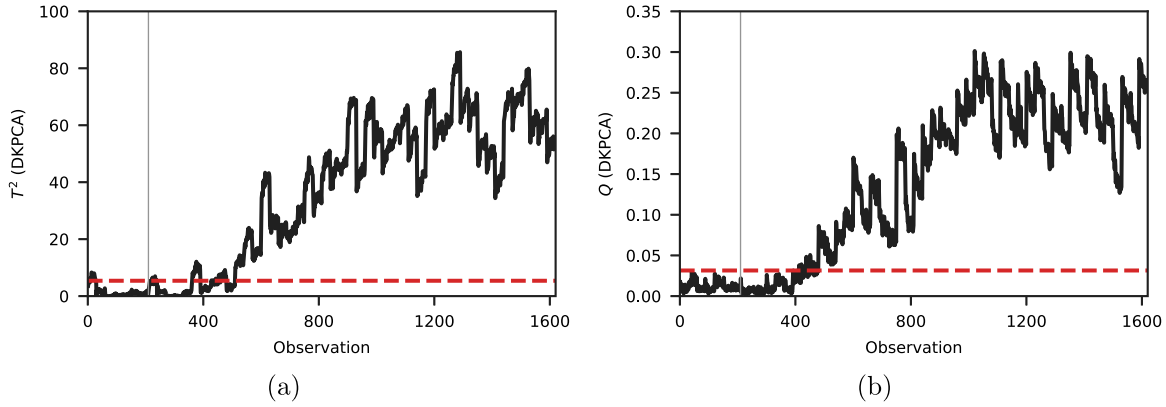

Fig. 20. ContCarSim case study: (a) T^2 statistic and (b) Q statistic for DKPCA applied to Fault 1. The dashed red line is the χ^2 control limit. The fault occurs at observation 200 (vertical line).

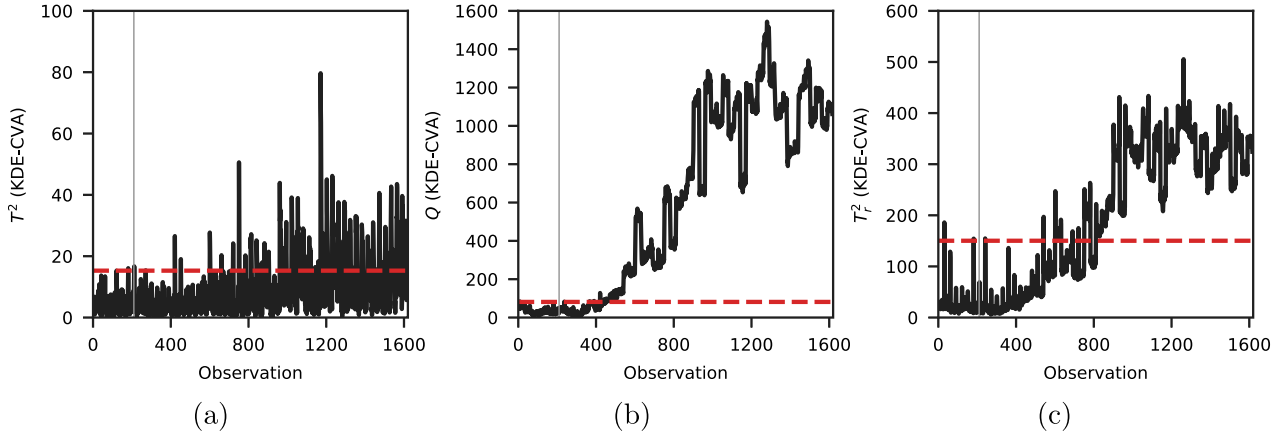

Fig. 21. ContCarSim case study: (a) T^2 statistic, (b) Q statistic, and (c) T^2 statistic, for KDE-CVA applied to Fault 1. The dashed red line is the χ^2 control limit. The fault occurs at observation 200 (vertical line).

Table 8
TEP case study: Type II error rates for all detectable faults for all methods not considering dependent variables separately. Cases where DPCA performs best are in bold.

Fault #	PCA	DPCA	KPCA	DKPCA	SVDD
1	0.004	**0.001**	1.000	0.098	0.003
2	0.014	**0.013**	0.986	0.936	0.014
4	0.063	**0.009**	0.935	0.838	0.029
5	0.606	0.578	0.927	0.630	0.551
6	0.000	**0.000**	1.000	0.966	0.000
7	0.000	**0.000**	0.399	0.338	0.000
8	0.013	0.011	0.899	0.117	0.006
10	0.345	0.320	0.770	0.404	0.316
11	0.273	**0.207**	0.849	0.721	0.228
12	0.006	**0.003**	0.896	0.142	0.004
13	0.040	**0.036**	0.990	0.467	0.041
14	0.000	**0.000**	0.957	0.058	0.000
16	0.419	**0.350**	0.862	0.484	0.352
17	0.068	**0.058**	0.966	0.576	0.064
18	0.086	0.084	0.995	0.934	0.075
19	0.799	**0.737**	0.972	0.906	0.738
20	0.333	0.292	0.931	0.475	0.273
21	0.539	0.503	0.947	0.619	0.500

Table 9
TEP case study: Type II error rates for all detectable faults for all methods considering dependent variables separately. Cases where CVA performs best are in bold.

Fault #	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
1	0.001	0.000	0.008	0.006	**0.000**	0.003
2	0.015	0.013	0.021	0.018	**0.008**	0.014
4	0.019	0.000	0.835	0.809	**0.000**	0.000
5	0.616	0.565	0.702	0.639	**0.000**	0.000
6	0.000	0.000	0.004	0.004	**0.000**	0.000
7	0.000	0.000	0.029	0.000	**0.000**	0.000
8	0.010	0.010	0.035	0.025	**0.010**	0.021
10	0.380	0.294	0.514	0.424	**0.073**	0.111
11	0.261	0.140	0.615	0.576	**0.127**	0.235
12	0.006	0.000	0.024	0.006	**0.000**	0.000
13	0.045	0.036	0.056	0.054	0.043	0.048
14	0.000	0.000	0.005	0.000	**0.000**	0.000
16	0.439	0.296	0.674	0.487	**0.041**	0.082
17	0.061	0.040	0.211	0.157	**0.025**	0.039
18	0.080	0.073	0.098	0.088	0.082	0.098
19	0.806	0.622	0.986	0.950	**0.044**	0.097
20	0.365	0.276	0.583	0.474	**0.083**	0.092
21	0.514	0.427	0.620	0.496	**0.302**	0.391

Table 10
ContCarSim case study: Type I and II error rates for Fault 1 for all methods not considering dependent variables separately.

	PCA	DPCA	KPCA	DKPCA	SVDD
Type I error rate	0.095	0.114	0.090	0.100	0.033
Type II error rate	0.133	0.123	0.143	0.124	0.129

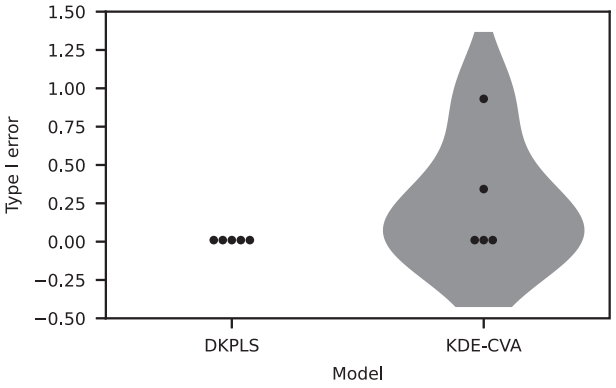

Fig. 22. Metal Etch Process case study: Violin plot of the validation errors for KDE-CVA and DKPLS. Each black dot marks the validation error for one of the folds.

- According to the "variables" criterion, the dataset is deemed nonlinear with a fraction of variables involved in nonlinear relationships equal to 0.4737 (9 nonlinear variables out of 19) in the case without dependent variables, and equal to 0.4444 (8 nonlinear variables out of 18) in the case with dependent variables. All approaches to deflate the maximal correlation coefficient yield essentially the same results, yet with some minor variations in the percentage of nonlinear variables.
- According to the "variables" criterion, the dataset is deemed dynamic with a fraction of dynamic variables equal to 0.647 (11 dynamic variables out of 17 non-categorical variables) in the case without dependent variables, and equal to 0.625 (10 dynamic variables out of 16 non-categorical variables) when output variables are considered.

The data interrogation criteria of SPAfPM indicate that dynamics and nonlinearity are present in the NOC data. Similar to the previous case studies, both cases with and without dependent variables separately are considered. In the former scenario, DKPCA would usually be selected according to the Fig. 2. However, due to the presence of discrete variables, SPAfPM recommends SVDD as the most suitable algorithm (see discussion in the Supplementary Material). k-fold cross-validation with 5 folds and 10 repeats is used to determine the hyperparameters, which are $C = 0.2$ and the radial basis function kernel with $\sigma = 50$. In the case that dependent variables are considered separately, DKPLS and KDE-CVA are recommended as the most suitable algorithms based on the found characteristics according to the data analytics triangle for fault detection shown in Fig. 2. Cross-validation for dynamics with 5 folds is used for model selection and hyperparameter tuning. Based on the error distribution reported in Fig. 22, DKPLS performs better overall by having a consistent Type I error of 0.010, whereas KDE-CVA has a larger error for some of the folds. The best model appears to be DKPLS, whereas SPAfPM recommends both DKPLS and KDE-CVA as the suitable algorithms (see the discussion in the Supplementary Material). The hyperparameter tuning for DKPLS results in a lag $h = 1$, a number of latent variables $a = 1$, plus a polynomial kernel with $c_0 = 1$, $d = 3$, and $\gamma = 0.0004$. The hyperparameters for KDE-CVA yield a lag $h = 1$, a memory order $a = 1$, and a kernel bandwidth scaling factors of $\xi_{T^2} = \xi_{T_r^2} = 1$ and $\xi_Q = 1$ for T^2 (and T_r^2) and Q, respectively.

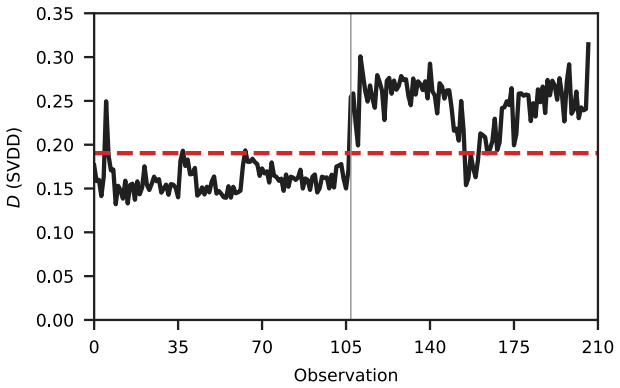

Fig. 23. Metal Etch Process case study: The distance statistic for SVDD applied to Fault 10. The dashed red line is the χ^2 control limit. The fault occurs at observation 107 (vertical line).

Fault 10 is discussed as an example in this case study. If no dependent variables are used, SVDD achieves a Type I error of 0.028 and a Type II error of 0.071 for Fault 10. An overview of the different algorithms not considering dependent variables separately is shown in Table 12. The distance statistic for SVDD is shown in Fig. 23. The performance of the recommended algorithm is overall very good. SVDD shows good performance for the Type I error and performs significantly better than all other algorithms for the Type II error. Very similar results are observed for Faults 1 and 16. SVDD significantly outperforms the other algorithms for the Type II error. The Type I error is consistent over all faults as the initial data before the fault occurs are the same.

In the case where dependent variables are considered separately, KDE-CVA outperforms DKPLS. KDE-CVA results in a Type I error of 0.028 and a Type II error of 0.000 for Fault 10. This result is the best, and equivalent to CVA. An overview of the different algorithms considering dependent variables separately is shown in Table 13. The T^2, T_r^2, and Q statistics for KDE-CVA are shown in Fig. 24. The performance of KDE-CVA is overall very good for both Type I and II errors. KDE-CVA performs second best on Fault 1 and best on Faults 10 and 16.

5. Conclusions

An automated approach is proposed for the selection of data-driven modeling methods for fault detection. The approach interrogates the given dataset for different characteristics, which are the presence of dependent variables, non-normality of the data distribution, presence of nonlinear correlation among variables, and dynamics in the data. The presence of dependent variables is a choice that requires process knowledge and is left as a decision to the user of the software. To determine the other characteristics, a rigorous data interrogation procedure was designed and validated in a multitude of different Monte Carlo simulations. This characteristic analysis, in combination with a cross-validation procedure, constitute the backbone of the automation mechanism of the proposed framework to select the most suitable fault detection algorithm with a set of optimized hyperparameters for the given dataset.

The proposed smart data analytics approach for fault detection is applied to four case studies, including the Tennessee Eastman Process, an established benchmark for process monitoring systems. Based on an analysis of the normal operating condition data available for such a case study, the framework selected DPCA in the case that dependent variables are not considered separately, and CVA in the case that dependent variables are considered separately. In terms of the Type II error rate, DPCA was the best performing algorithm not considering dependent variables for 12 out of the 18 faults used for testing, and CVA was the best performing algorithm for 16 out of 18 faults when

Table 11
ContCarSim case study: Type I and II error rates for Fault 1 for all methods considering dependent variables separately.

	PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
Type I error rate	0.033	0.029	0.029	0.038	0.043	0.033
Type II error rate	0.134	0.122	0.162	0.125	0.147	0.158

Table 12
Metal Etch case study: Type I and II error rates for Faults 1, 10, and 16 for all methods not considering dependent variables separately.

Fault #		PCA	DPCA	KPCA	DKPCA	SVDD
1	Type I error rate	0.019	0.000	0.000	0.000	0.028
	Type II error rate	0.981	0.980	1.000	1.000	0.139
10	Type I error rate	0.019	0.009	0.000	0.000	0.028
	Type II error rate	0.600	0.354	1.000	1.000	0.071
16	Type I error rate	0.019	0.009	0.000	0.000	0.028
	Type II error rate	0.860	0.616	1.000	1.000	0.041

Table 13
Metal Etch Process case study: Type I and Type II error rates for Faults 1, 10, and 16 for all methods considering dependent variables separately.

Fault #		PLS	DPLS	KPLS	DKPLS	CVA	KDE-CVA
1	Type I error rate	0.019	0.009	0.000	0.009	0.037	0.028
	Type II error rate	0.990	0.971	1.000	0.990	0.530	0.828
10	Type I error rate	0.019	0.019	0.000	0.009	0.037	0.028
	Type II error rate	0.670	0.374	1.000	0.673	0.000	0.000
16	Type I error rate	0.019	0.019	0.000	0.019	0.037	0.028
	Type II error rate	0.830	0.616	1.000	0.867	0.000	0.000

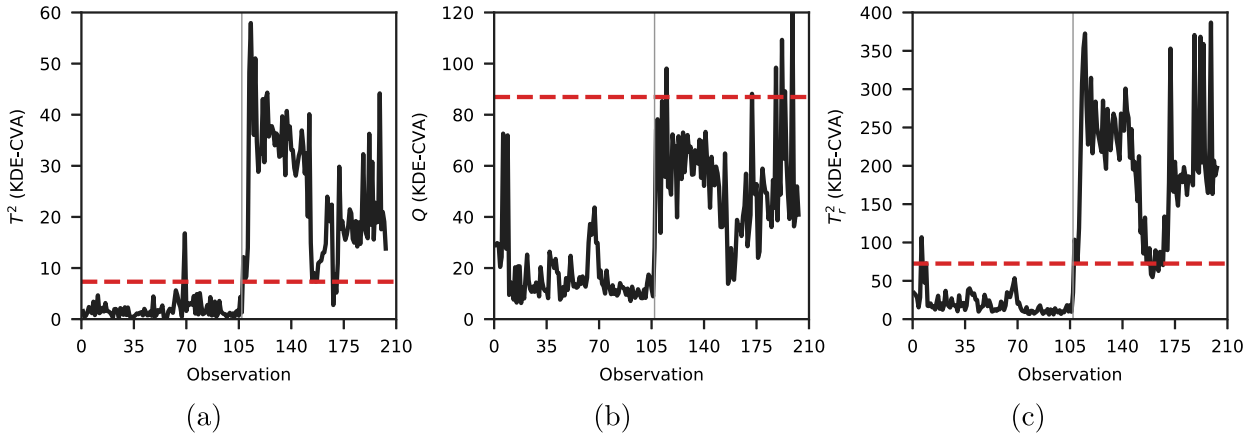

Fig. 24. Metal Etch Process case study: (a) T^2 statistic, (b) Q statistic, and (c) T^2 statistic, for KDE-CVA applied to Fault 10. The dashed red line is the χ^2 control limit. The fault occurs at observation 107 (vertical line).

dependent variables are considered. This performance is strong, as it is known that there is not one algorithm that does best on any possible scenario; the software successfully determined the algorithm that performs best for most faults. Three other relevant case studies are assessed, with similar performance to the Tennessee Eastman Process. Overall, the proposed approach successfully suggests the most suitable algorithm and determines the optimal set of hyper parameters based on a rigorous cross-validation procedure in a fully automated way requiring no fault detection expert knowledge by the user.

Acronyms

ACF AutoCorrelation Function

API Active Pharmaceutical Ingredient

AutoML Automatic Machine Learning

CVA Canonical Variate Analysis

DKPCA Dynamic Kernel Principal Component Analysis

DKPLS Dynamic Kernel Partial Least-Squares

DPCA Dynamic Principal Component Analysis

DPLS Dynamic Partial Least-Squares

KDE Kernel Density Estimation

KPCA Kernel Principal Component Analysis

KPLS Kernel Partial Least-Squares

NOC Normal Operating Conditions

OCC One-Class Classification

PCA Principal Component Analysis

PLS Partial Least-Squares

RBF Radial Basis Function

SPA Smart Process Analytics

SPAfPM Smart Process Analytics for Process Monitoring

SVDD Support Vector Data Description

TEP Tennessee Eastman Process

CRediT authorship contribution statement

Fabian Mohr: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Elia Arnese-Feffin:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Massimliano Barolo:** Supervision, Funding acquisition. **Richard D. Braatz:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financially supported by the U.S. Food and Drug Administration, United States, Contract No. 75F40121C00090. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the financial sponsor.

This work has been partially funded by Fondazione Ing. Aldo Gini (University of Padova) through a Scholarship awarded to E.A.F, Italy. for his research period at MIT. The authors also gratefully acknowledge financial support by Novamont S.p.A., CARIPARO Foundation, UniSmart Foundation, and IntesaSanPaolo through the PhD Scholarship of E.A.F.

Data availability

The code of SPAfPM used to obtain the results discussed in this paper and data for each one of the case studies discussed herein are available at the GitHub repository of SPAfPM: https://github.com/EliaAF/SmartProcessAnalyticsforProcessMonitoring

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compchemeng.2024.108918.

References

Abid, A., Khan, M.T., Iqbal, J., 2021. A review on fault detection and diagnosis techniques: basics and beyond. Artif. Intell. Rev. 54, 3639–3664.

Allen, D.M., 1974. The relationship between variable selection and data agumentation and a method for prediction. Technometrics 16, 125–127.

Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection. Stat. Surv. 4, 40–79.

Arnese Feffin, E., 2023. Industry 4.0 in industrial biorefineries: improving process operations by data-driven and hybrid modeling (Ph.D. thesis). University of Padova.

Azamfar, M., Li, X., Lee, J., 2020. Deep learning-based domain adaptation method for fault diagnosis in semiconductor manufacturing. IEEE Trans. Semicond. Manuf. 33, 445–453.

Bergmeir, C., Benítez, J.M., 2012. On the use of cross-validation for time series predictor evaluation. Inform. Sci. 191, 192–213.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition, fourth ed. Clarendon Press.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2016. Time Series Analysis, fifth ed. Wiley.

Breiman, L., Friedman, J.H., 1985. Estimating optimal transformations for multiple regression and correlation. J. Amer. Statist. Assoc. 80, 580–598.

Brereton, R.G., 2011. One-class classifiers. J. Chemometr. 25, 225–246.

Burman, P., 1989. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. Biometrika 76, 503–514.

Camacho, J., Ferrer, A., 2012. Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Theoretical aspects. J. Chemometr. 26, 361–373.

Camacho, J., Ferrer, A., 2014. Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Practical aspects. Chemometr. Intell. Lab. Syst. 131, 37–50.

Camacho, J., Pérez-Villegas, A., Garciá-Teodoro, P., Maciá-Fernández, G., 2016. PCA-based multivariate statistical network monitoring for anomaly detection. Comput. Secur. 59, 118–137.

Camacho, J., Picó, J., 2006a. Multi-phase principal component analysis for batch processes modelling. Chemometr. Intell. Lab. Syst. 81, 136–172.

Camacho, J., Picó, J., 2006b. Online monitoring of batch processes using multi-phase principal component analysis. J. Process Control 16, 1021–1035.

Camacho, J., Picó, J., Ferrer, A., 2008. Bilinear modelling of batch processes. Part I: Theoretical discussion. J. Chemometr. 22, 299–308.

Camacho, J., Picó, J., Ferrer, A., 2009. The best approaches in the on-line monitoring of batch processes based on PCA: Does the modelling structure matter? Anal. Chim. Acta 642, 59–68.

Chen, T., Zhang, J., 2010. On-line multivariate statistical monitoring of batch processes using Gaussian mixture model. Comput. Chem. Eng. 34, 500–507.

Cherry, G.A., Qin, S.J., 2007. Monitoring non-normal data with principal component analysis and adaptive density estimation. pp. 352–359.

Chiang, L.H., Russell, E.L., Braatz, R.D., 2001. Fault Detection and Diagnosis in Industrial Systems, first ed. Springer.

Cho, J.H., Lee, J.M., Choi, S.W., Lee, D., Lee, I.B., 2005. Fault identification for process monitoring using kernel principal component analysis. Chem. Eng. Sci. 60, 279–288.

Choi, S.W., Lee, I.B., 2004. Nonlinear dynamic process monitoring based on dynamic kernel PCA. Chem. Eng. Sci. 59, 5897–5908.

Choi, S.W., Lee, C., Lee, J.M., Park, J.H., Lee, I.B., 2005. Fault detection and identification of nonlinear processes based on kernel PCA. Chemometr. Intell. Lab. Syst. 75, 55–67.

Cui, P., Li, J., Wang, G., 2008. Improved kernel principal component analysis for fault detection. Expert Syst. Appl. 34, 1210–1219.

Destro, F., 2022. Continuous Carousel Simulator Code. URL https://github.com/CryPTSyS/ContCarSim.

Destro, F., Hur, I., Wang, V., Abdi, M., Feng, X., Wood, E., Coleman, S., Firth, P., Barton, A., Barolo, M., Nagy, Z.K., 2021. Mathematical modeling and digital design of an intensified filtration-washing-drying unit for pharmaceutical continuous manufacturing. Chem. Eng. Sci. 224, 116803.

Destro, F., Nagy, Z.K., Barolo, M., 2022. A benchmark simulator for quality-by-design and quality-by-control studies in continuous pharmaceutical manufacturing – intensified filtration-drying of crystallization slurries. Comput. Chem. Eng. 163, 107809.

Dong, D., Mcavoy, T.J., 1996. Nonlinear principal component analysis - based on principal curves and neural networks. Comput. Chem. Eng. 20, 65–78.

Downs, J.J., Vogel, E.F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17, 245–255.

Du, X., 2019. Fault detection using bispectral features and one-class classifiers. J. Process Control 83, 1–10.

Eigenvector Research, Inc., 2024. Eigenvector Research Datasets. URL https://eigenvector.com/resources/data-sets/.

Feurer, M., Klein, A., Springenberg, J.T., Blum, M., Hutter, F., 2015. Efficient and robust automated machine learning. In: Proceeding of the 29th Annual Conference on Neural Information Processing Systems, NIPS 2015. pp. 2962–2970.

Filzmoser, P., Liebmann, B., Varmuza, K., 2009. Repeated double cross validation. J. Chemometr. 23, 160–171.

Geladi, P., Kowalski, B.R., 1986. Partial least-squares regression: A tutorial, vol. 185. pp. 1–17.

Goeman, J.J., Solari, A., 2014. Multiple hypothesis testing in genomics. Stat. Med. 33, 1946–1978.

Goodlin, B.E., Boning, D.S., Sawin, H.H., Wise, B.M., 2003. Simultaneous fault detection and classification for semiconductor manufacturing tools. J. Electrochem. Soc. 150, G778–G784.

H2O AI, 2024. H2O AutoML. URL https://github.com/h2oai/h2o-3.

Hastie, T., Tibshirani, R., Friedman, J.H., 2009. The Elements of Statistical Learning, second ed. Springer.

He, Q.P., Wang, J., 2011. Statistics pattern analysis: A new process monitoring framework and its application to semiconductor batch processes. AIChE J. 57, 107–121.
Henze, N., Zirkler, B., 1990. A class of invariant consistent tests for multivariate normality. Comm. Statist. Theory Methods 19, 3595–3617.
Hochberg, Y., 1988. A sharper Bonferroni procedure for multiple tests of significance. Biometrika 75, 800–802.
Hutter, F., Kotthoff, L., Vanschoren, J., 2019. Automated Machine Learning. Spinger.
Isermann, R., 1984. Process fault detection based on modeling and estimation methods—A survey. Automatica 20, 307–404.
Isermann, R., 1994. Integration of fault detection and diagnosis methods. IFAC Proc. Vol. 27, 575–590.
Isermann, R., 1997. Supervision, fault-detection and fault-diagnosis methods — An introduction. Control Eng. Pract. 5, 639–652.
Isermann, R., 2005. Model-based fault-detection and diagnosis–status and applications. Annu. Rev. Control 29, 71–85.
Jackson, J.E., 1959. Quality conrol methods for several related variables. Technometrics 21, 341–349.
Jackson, J.E., Mudholkar, G.S., 1979. Control procedures for residuals associated with principal component analysis. Technometrics 21, 341–349.
Jia, Q., Zhang, Y., 2016. Quality-related fault detection approach based on dynamic kernel partial least squares. Chem. Eng. Res. Des. 106, 242–252.
Jiang, B., Zhu, X., Huang, D., Paulson, J.A., Braatz, R.D., 2015. A combined canonical variate analysis and Fisher discriminant analysis (CVA–FDA) approach for fault diagnosis. Comput. Chem. Eng. 77, 1–9.
Jiao, J., Yu, H., Wang, G., 2015. A quality-related fault detection approach based on dynamic least squares for process monitoring. IEEE Trans. Ind. Electron. 63, 2625–2632.
Jin, H., Song, Q., Hu, X., 2019. Auto-keras: An efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1946–1956.
Korkmaz, S., Goksuluk, D., Zararsiz, G., 2014. MVN: An R package for assessing multivariate normality. R J. 6, 151–162.
Kotthoff, L., Thornton, C., Hoos, H.H., Hutter, F., Leyton-Brown, K., 2017. Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA. J. Mach. Learn. Res. 18, 1–5.
Ku, W., Storer, R.H., Georgakis, C., 1995. Disturbance detection and isolation by dynamic principal component analysis. Chemometr. Intell. Lab. Syst. 30, 179–196.
Larimore, W.E., 1990. Canonical variate analysis in identification, filtering, and adaptive control. In: Proceedings of the IEEE Conference on Decision and Control. pp. 596–604.
Le, T.T., Fu, W., Moore, J.H., 2020. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics 36, 250–256.
Li, G., Liu, B., Qin, S.J., Zhou, D., 2011. Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: The dynamic T-PLS approach. IEEE Trans. Neural Netw. 22, 1622–2271.
Ljung, G.M., Box, G.E.P., 1978. On a measure of lack of fit in time series models. Biometrika 65, 279–303.
Lu, N., Yao, Y., Gao, F., Wang, F., 2005. Two-dimensional dynamic PCA for batch process monitoring. AIChE J. 51, 3200–3204.
Lv, F., Wen, C., Liu, M., Bao, Z., 2018. Higher-order correlation-based multivariate statistical process monitoring. J. Chemometr. 32, e3033.
Makridakis, S., 1990. Sliding simulation: A new approach to time-series forecasting. Manage. Sci. 36, 505–512.
Mardia, K.V., 1970. Measures of multivariate skewness and kurtosis with applications. Biometrika 57, 519–530.
Md Nor, N., Che Hassan, C.R., Hussain, M.A., 2020. A review of data-driven fault detection and diagnosis methods: applications in chemical process systems. Rev. Chem. Eng. 36, 513–553.
Mecklin, C.J., Mundfrom, D.J., 2005. A Monte Carlo comparison of the type I and type II error rates of tests of multivariate normality. J. Stat. Comput. Simul. 75, 93–107.
MLJAR, 2024. MLJAR. URL https://github.com/mljar/mljar-supervised.
Mohr, F., 2024. Smart Data Analytics for Manufacturing Processes (Ph.D. thesis). Massachusetts Institute of Technology.
Mohr, F., Sun, W., Braatz, R.D., 2022. Smart data analytics for supervised classification.
Montgomery, D.C., Runger, G.C., 2018. Applied Statistics and Probability for Engineers, Seventh ed. Wiley.
Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B., 2001. An introduction to kernel-based learning algorithms. IEEE Trans. Neural Netw. 12, 181–201.
Nadon, R., Shoemaker, J., 2002. Statistical issues with microarrays: processing and analysis. TRENDS Genet. 18, 265–271.
Nomikos, P., MacGregor, J.F., 1994. Monitoring batch processes using multiway principal component analysis. AIChE J. 40, 1361–1375.
Nomikos, P., MacGregor, J.F., 1995. Multivariate processes SPC charts for monitoring batch processes. Technometrics 37, 41–59.
Odiowei, P.P., Cao, Y., 2010. Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations. IEEE Trans. Ind. Inform. 6, 36–45.
Oliveri, P., López, M.I., Casolino, M.C., Ruisánchez, I., Callao, M.P., Medini, L., Lanteri, S., 2014. Partial least squares density modeling (PLS-DM) – a new class-modeling strategy applied to the authentication of olives in brine by near-infrared spectroscopy. Anal. Chim. Acta 851, 30–36.
Paluš, M., Dvořák, I., 1992. Singular-value decomposition in attractor reconstruction: Pitfalls and precautions. Physica D 55, 221–234.
Parzen, E., 1962. On the estimation of probability density functions and mode. Ann. Math. Stat. 33, 1065–1076.
Qin, S.J., 2003. Statistical process monitoring: Basics and beyond. J. Chemometr. 17, 480–502.
Raich, A., Çinar, A., 1996. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. AIChE J. 42, 995–1009.
Ramaker, H.J., Van Sprang, E.N.M., Westerhuis, J.A., Gurden, S.P., Smilde, A.K., 2006. Performance assessment and improvement of control charts for statistical batch process monitoring. Stat. Neerl. 60, 339–360.
Rato, T.J., Reis, M.S., 2013. Fault detection in the Tennessee eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR). Chemometr. Intell. Lab. Syst. 125, 101–108.
Reis, M.S., Rendall, R., Rato, T.J., Martins, C., Delgado, P., 2021. Improving the sensitivity of statistical process monitoring of manifolds embedded in high-dimensional spaces: the truncated-q statistic. Chemometr. Intell. Lab. Syst. 215 (104369).
Rényi, A., 1959. On measures of dependence. Acta Math. Hungar. 10, 441–451.
Ricker, N.L., 1988. The use of biased least-squares estimators for parameters in discrete-time pulse-response models. Ind. Eng. Chem. Res. 27, 343–350.
Rodionova, O.Y., Oliveri, P., Pomerantsev, A.L., 2016. Rigorous and compliant approaches to one-class classification. Chemometr. Intell. Lab. Syst. 159, 89–96.
Rosenblatt, M., 1956. Remarks on some nonparametric estimates of a density function. Ann. Math. Stat. 27, 832–837.
Rosipal, R., Trejo, L.J., 2001. Kernel partial least squares regression in reproducing kernel Hilbert space. J. Mach. Learn. Res. 2, 97–123.
Royston, J.P., 1983. Some techniques for assessing multivariate normality based on the shapiro-wilk w. Appl. Stat. 32, 121–133.
Russell, E.L., Chiang, L.H., Braatz, R.D., 2000. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemometr. Intell. Lab. Syst. 51, 81–93.
Salesforce, 2020. TransmogrifAI. URL https://github.com/salesforce/TransmogrifAI.
Schölkopf, B., Smola, A., Müller, K.R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10, 1299–1319.
Stone, M., 1974. Cross-validatory choice and assessment of statistical predictions. J. R. Stat. Soc. Ser. B Stat. Methodol. 36, 111–133.
Sun, W., 2020a. Smart process analytics for predictive modeling. URL https://github.com/vickysun5/SmartProcessAnalytics.
Sun, W., 2020b. Advanced Process Data Analytics (Ph.D. thesis). Massachussets Institute of Technology.
Sun, W., Braatz, R.D., 2021. Smart process analytics for predictive modeling. Comput. Chem. Eng. 144, 107134.
Tax, D.M.J., Duin, R.P.W., 1999. Support vector domain description. Pattern Recognit. Lett. 20, 1191–1199.
Tax, D.M.J., Duin, R.P.W., 2004. Support vector data description. Mach. Learn. 54, 45–66.
Thissen, U., Melssen, W.J., Buydens, L.M.C., 2001. Nonlinear process monitoring using bottle-neck neural networks. Anal. Chim. Acta 446, 369–381.
Tibshirani, R., 1988. Estimating transformations for regression via additivity and variance stabilization. J. Am. Statist. Assoc. 83, 394–405.
Tien, D.X., Lim, K.W., Jun, L., 2004. Comparative study of PCA approaches in process monitoring and fault detection. In: 30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004.
Tien, D.X., Lim, K.W., Jun, L., 2012. Comparative study of PCA approaches in process monitoring and fault detection. In: Second International Conference on Intelligent System Design and Engineering Application.
Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., 2003a. A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies. Comput. Chem. Eng. 27, 313–326.
Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003b. A review of process fault detection and diagnosis. Part III: Process history based methods. Comput. Chem. Eng. 27, 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003c. A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. Comput. Chem. Eng. 27, 293–311.

Wang, L., Shi, H., 2014. Improved kernel PLS-based fault detection approach for nonlinear chemical processes. Chin. J. Chem. Eng. 22, 657–663.

Wang, H., Yao, M., 2015. Chemometrics and intelligent laboratory systems fault detection of batch processes based on multivariate functional kernel principal component analysis. Chemometr. Intell. Lab. Syst. 149, 78–89.

Wise, B.M., Gallagher, N.B., 1996. The process chemometrics approach to process monitoring and fault detection. J. Process Control 6, 329–348.

Wise, B.M., Gallagher, N.B., Butler, S.W., White, D.D., Barna, G.G., 1999. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. J. Chemometr. 13, 379–396.

Wold, S., 1987. Principal component analysis. Chemometr. Intell. Lab. Syst. 2, 37–52.

Wold, S., Sjöström, M., Eriksson, L., 2001. PLS-regression: A basic tool of chemometrics. Chemometr. Intell. Lab. Syst. 58, 109–130.

Xu, L., Oja, E., Suen, C.Y., 1992. Modified hebbian learning for curve and surface fitting. Neural Netw. 5, 441–457.

Yao, Y., Gao, F., 2007. Batch process monitoring in score space of two-dimensional dynamic principal component analysis (PCA). Ind. Eng. Chem. Res. 46, 8033–8043.

Yin, S., Ding, S.X., Zhang, P., Hagahni, A., Naik, A., 2011. Study on modifications of PLS approach for process monitoring. IFAC Proc. Vol. 44, 12389–12394.

Yoon, S., MacGregor, J.F., 2004. Principal-component analysis of multiscale data for process monitoring and fault diagnosis. AIChE J. 50, 2891–2903.

Zhang, Y., Qin, S.J., 2008. Improved nonlinear fault detection technique and statistical analysis. AIChE J. 54, 3208–3220.

Zhu, Q., Liu, Q., Qin, S.J., 2016. Concurrent canonical correlation analysis modeling for quality-relavant monitoring. IFAC-PapersOnLine 49, 1044–1049.