Journal of Process Control 135 (2024) 103178

Contents lists available at ScienceDirect

Journal of Process Control

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

Data-based decomposition plant for decentralized monitoring schemes: A comparative study

M.J. Fuente, M. Galende-Hernández, G.I. Sainz-Palmero [*]

Department of System Engineering and Automatic Control, Industrial Engineering School, Universidad de Valladolid, Paseo Prado de la Magdalena 3-5, 47011 Valladolid, Spain

ARTICLE INFO

Keywords:
Fault detection
Canonical Variate Analysis
Regression
Correlation
Mutual information
Clustering
Decentralized process monitoring
Bayesian Inference

ABSTRACT

The complexity of the industrial processes, large-scale plants and the massive use of distributed control systems and sensors are challenges which open ways for alternative monitoring systems. The decentralized monitoring methods are one option to deal with these complex challenges. These methods are based on process decomposition, i.e., dividing the plant variables into blocks, and building statistical data models for every block to perform local monitoring. After that, the local monitoring results are integrated through a decision fusion algorithm for a global output concerning the process. However, decentralized process monitoring has to deal with a critical issue: a proper process decomposition, or block division, using only available data. Knowledge of the plant is rarely available, so data-driven approaches can help to manage this issue. Moreover, this is the first and key step to developing decentralized monitoring models and several alternative approaches are available. In this work a comparative study is carried out regarding decentralized fault monitoring methods, comparing several alternative proposals for process decomposition based on data. These methods are based on information theory, regression and clustering, and are compared in terms of their monitoring performance. When the blocks are obtained, CVA (Canonical Variate Analysis) based local dynamic monitors are set up to characterize the local process behavior, while also considering the dynamic nature of the industrial plants. Finally, the Bayesian Inference Index (BII) is implemented, based on these local monitoring, to achieve a global outcome regarding fault detection for the whole process. To further compare their performance from the application viewpoint, the Tennessee Eastman (TE) process, a well-known industrial benchmark, is used to illustrate the efficiencies of all the discussed methods. So, a systematically comparison have been carried out involving different data-driven methods for process decomposition to implement a decentralized monitoring scheme. The results are focused on providing a reference for practitioners as guidelines for successful decentralized monitoring strategies.

1. Introduction

Process monitoring for complex large-scale industrial plants is an effective way to improve safety, prevent damage to equipment and maintain normal production by detecting anomalies and diagnosing their root cause. Nowadays, the application of distributed control systems and modern measurement techniques is widespread, which can make large amount of data about the plant available. Due to all this, data-driven process monitoring has become more popular and usual, particularly Multivariate Statistical Process Monitoring (MSPM) which has been intensively researched, obtaining impressive progress over the last decades [1–4].

These techniques, including Principal Component Analysis (PCA), Partial Least Squares (PLS) or Independent Component Analysis (ICA), effectively extract the underlying characteristics from historical data and drive normal operation models by accommodating acceptable variations, and detecting abnormal conditions using statistical metrics [5–7]. However, it is accepted that these traditional methods may not be the best for complex industrial plants with their dynamic and non-linear nature. In order to manage these problems, complementary approaches are also included, such as Canonical Variate Analysis (CVA) and Dynamical PCA (DPCA) for dynamic process monitoring, or kernel techniques (KPCA, KPLS and KICA) for non-linear processes [8–14].

In order to face the challenge regarding wide-plant and large-scale process monitoring some proposals address the challenge through decentralized strategies. These approaches introduce some advantages: first of all, the process decomposition into blocks can reduce the computational loadings since each block employs a smaller set of relevant

[*] Corresponding author.
E-mail addresses: mariajesus.fuente@uva.es (M.J. Fuente), marta.galende@uva.es (M. Galende-Hernández), gregorioismael.sainz@uva.es (G.I. Sainz-Palmero).

https://doi.org/10.1016/j.jprocont.2024.103178
Received 18 October 2023; Received in revised form 13 January 2024; Accepted 4 February 2024
Available online 9 February 2024

variables to process. Secondly, by dividing the plant into multiple blocks, it is possible to detect fault events in a distributed manner, and the most responsible faulty sections and faulty variables can be isolated. Finally, through building multiple blocks the robustness of the whole statistic model can be enhanced for maintenance process, because regular maintenance and updating of some blocks has no impact on other blocks [15].

A common key issue of these decentralized strategies is the decomposition of the plant variables into blocks to be processed by a local monitoring model and, finally, a central processor to collect all local outcomes and fuse them for a global decision regarding the normal or faulty state of the whole process. However, the main challenge of this strategy is the proper process decomposition, i.e., to divide the variables into overlapping or disjoint blocks, which is the first and key step for almost all decentralized monitoring strategies.

The traditional decomposition methods are usually based on prior knowledge or process topology [16–18]; however, this knowledge is not usually available, especially for complex industrial plants. So, other alternatives, such as data-driven approaches, are interesting for dealing with this critical issue: such alternatives would include [19] sparse PCA; or [20,21] using PCA to built the blocks in the different principal component directions, or [22–25] using mutual information and correlation between the variables, respectively; or [26] first divides the variables into Gaussian and non-Gaussian subspaces by Jarque–Bera detection method, and after that uses the mutual information (MI) in both subspaces to determine quality-relevant and quality-irrelevant variables to obtain the sub-blocks.

Other authors, as [27,28] use the minimal redundancy and maximal relevance (mRMR) method to carry out the decomposition; also similarly the method proposed by [29] is based on relevance and redundancy (RRVS) taking into account the strong dynamic relation between the variables of the process. We have also proposed several of our own techniques to decompose the plant by linear regression (Sparse PLS) [30], LASSO, Elastic-net and non-linear regression by neural networks [31], or based on clustering [32]. Recently, [33] combine the mechanism knowledge and data analysis to decentralize the plant, i.e., the plant wide process is preliminary map as an undirected graph corresponding to the mechanism knowledge and process structure which divides the plant into blocks and secondly the authors use mutual information based Louvain algorithm (MI-Louvain) to fine decompose the process into reasonable sub-blocks, considering the correlation among variables and the graph structure.

Others authors use causal networks that holds a large amount of information on how process variables relate each other. For example, in [34] the process variables are divide into modules, defined as communities, based on the analysis of this causal network through a community detection algorithm that evaluates the network topology and the density of associations between variables and then each module is monitored by a causal method called Sensitivity Enhancing Transformation (SET). Another idea based on networks is the cascaded monitoring network called MoniNet [35], that takes into consideration the local correlation between the different operation units in an industrial plant. In this method a convolutional operation for each variable is carried out to extract simultaneously temporal information. This reveals dynamic correlation on process data and spatial information, reflecting local characteristics within each operation unit. For each feature obtained, a sub-model is developed, and all the sub-models are integrated to generate a final monitoring model. The MoniNet model can be expanded to capture deeper information by adding more convolutional layers.

Also, [36] uses convolutional feature extraction as MoniNet, but in this method is firstly identified the causal relationship within the industrial process to analyze the interaction between different variables, and based on this graph structure, convolutional filters are designed to extract feature from directly related variables and their corresponding time lags. The obtained features of all process variables are combined into feature matrices based on which the sub-models are developed, using Autoencoders (AE) in this case.

Other perspective for the division of the variables of the wide-plant is the control-oriented decomposition for identifying subsystems within the plant each of which can be controlled effectively without affecting the other subsystems significantly [37]. This problem related to the classical problem of control structure design, i.e., the selection and pairing of manipulated input and controlled output variables, can be viewed from a network perspective, and corresponds to identify communities whose members interact strongly among them, yet are weakly coupled to the rest of the network members. So, this can be solve from the point of view of the graph and network theory [37]. But this kind of decomposition is out the scope of this paper.

All these proposals decompose the plant using only historical data to manage the shortage of knowledge. However, there is a lack of comparatives regarding the performance of these decomposition methods in terms of fault detection schemes in the bibliographic publications to rank them and know the best cases.

So, the main objective and contribution of this paper is focused on supplying a comparative performance study between different full driven-data methods for process decomposition into blocks to implement a decentralized monitoring scheme. The methods involved for this comparison are: based on linear regression, such as LASSO, Elastic net [31] or Sparse Partial Least Squares (SPLS) [30]; and based on non-linear regression using MLP neural networks [31]. The third kind of methods are based on information theory such as Sparse Principal Component Analysis (SPCA) [19], correlation analysis [23], mutual information [22], Detrended Cross-Correlation analysis (DCCA) or minimal redundancy and maximal relevance (mRMR) [28]; and finally, methods based on clustering [32]. All these methods are data driven proposals to divide the plant variables into blocks of variables. However, the majority of the distributed approaches use PCA for motioning each block, that only consider the static process variation, ignoring the dynamic characteristics of the industrial processes, so in this work for each block, a local dynamic multivariate statistical model, here based on Canonical Variate Analysis (CVA), is used for local monitoring. Finally, all the outcomes from the blocks are fused through the Bayesian inference strategy to provide an operation status of the whole plant.

The performance comparison between the different plant decomposition methods is based on the fault detection results using the following indexes: fault alarm rate (FAR), missed detection rate (MDR) and detection delay (DD).

On the other hand, another important index to consider is the complexity of the decomposition method, i.e., the number of blocks defined for every method (NB): some are fully decentralized approaches, [23,25,30], defining a block for each plant variable which can be unbearable for large scale plants. Therefore, to carry out this performance comparative the well-known Tennessee Eastman Process (TEP) benchmark is used.

In short, the main contribution of this work is to provide performance guidelines for data driven decomposition methods in decentralized monitoring strategies when facing large and complex process plants. Some of the block decomposition approaches included in this comparison have not been previously used for this goal in the known bibliography, as the DCCA method.

The rest of the paper is organized as follows. Section 2 provides some background knowledge about CVA and explains the different methods to perform the decentralization, while the probabilistic Bayesian fusion technique is also detailed here. Section 3 elaborates the decentralized fault detection method used in the paper. The results of the comparison between the different plant decomposition methods tested over the Tennessee Eastman Plant are summarized in Section 4, followed by the conclusions in Section 5.

2. Materials and methods

The relevant methods for the decentralized monitoring scheme are reviewed in this section. This is made up of three steps: first the methods used for the process decomposition based on data are described, followed by a brief review of CVA, which serves as the basis of the local fault detection method for each block, and finally the decision making to fuse all the results provided by each block, the Bayesian inference index (BII) is introduced.

2.1. Plant decomposition based on data

In this section, the different data-based techniques to decompose a large plant are briefly introduced. These methods are based on regression (linear: LASSO, Elastic net and SPLS and non linear: artificial neural networks); on information concerning the variables (Sparse PCA, correlation, mutual information, minimal redundancy and maximal relevance (mRMR) and Detrended Cross-Correlation analysis); and finally based on clustering.

LASSO method. Least Absolute Shrinkage and Selection Operator (LASSO) method [38] is a linear regression method, which penalizes the coefficients of the regression, making some of them zero, permitting a model simpler and more interpretable. This technique can be used to plant decomposition, using only the most relevant regressors, which have non-zero coefficients [31].

A linear model with m predictors: $X(t) = (x_1(t), x_2(t), \dots, x_m(t))$ and one response variable $y(t)$ can be expressed as follows:

$$Y = \beta \mathbf{X} + E \tag{1}$$

The LASSO method solves the problem as:

$$\hat{\beta}(\lambda) = \underset{\beta}{\operatorname{argmin}} \left(\frac{\| y - \beta \mathbf{X} \|_2^2}{n} + \lambda \| \beta \|_1 \right) \tag{2}$$

where $\lambda \geq 0$. This hyper-parameter λ controls the process: when its value increases, more coefficients are forced to be zero.

Elastic net method. The elastic net regression was introduced by [39] and can be seen as a compromise between Ridge and LASSO regression, i.e., it selects variables such as LASSO and shrinks the coefficients according to Ridge. So, the elastic net regression solves the optimization problem:

$$\hat{\beta}(\lambda, \delta) = \underset{\beta}{\operatorname{argmin}} \left(\frac{\| y - \beta \mathbf{X} \|_2^2}{n} + \lambda \left(\frac{1-\delta}{2} \| \beta \|_2^2 + \delta \| \beta \|_1 \right) \right) \tag{3}$$

The elastic net is more flexible, and for $\delta = 1$, it gives the LASSO solution and for $\delta = 0$ the Ridge regression is obtained. A frequent strategy is to assign a big value to the $L1$ penalization in order to get a lower number of predictors, i.e., putting a value of δ near to 1, and giving a little weight to the $L2$ regularization so as to provide some stability if some of the predictors are highly correlated. This technique can be used to perform feature selection, for the regression model of a variable x_i only the most relevant regressors are take into account, i.e., the variables x_j with a coefficient $\beta_j > len_i$ (see Eq. (1)) have to be in the same block as x_i with len_i a certain threshold [31].

SPLS method. The Sparse Partial Least Squares (SPLS) method [40] is also an improvement over the well known PLS method. This adjusts a linear model using least squares over new discovered features which are combinations of the original ones: $\mathbf{X}$ the predictor matrix $(n \times m)$ and $\mathbf{Y}$ the response matrix $(n \times p)$:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathbf{T}} + \mathbf{E} \tag{4}$$

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^{\mathbf{T}} + \mathbf{F} \tag{5}$$

where $\mathbf{T} = \mathbf{X}\mathbf{W}$ contains the first k terms of the latent variables or the score vectors, $\mathbf{W} = (w_1, w_2, \dots, w_k)$ is a matrix of direction vectors, $\mathbf{P}$ and $\mathbf{Q}$, respectively, are the loading vectors of the data matrices $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{E}$ and $\mathbf{F}$ are the residual terms of PLS. PLS has successively to calculate the vectors w that maximize the covariance between the explanatory variables $\mathbf{X}$ and the responses $\mathbf{Y}$, obtaining the regression model, with $\mathbf{B_{PLS}} = \mathbf{W}\mathbf{Q}$:

$$\mathbf{Y} = \mathbf{X}\mathbf{B_{PLS}} + \mathbf{F} \tag{6}$$

Now the objective of Sparse Partial Least Squares (SPLS) [40] is to obtain a number of coefficients equals to zero in vector $\mathbf{B_{PLS}}$, i.e., to assure sparsity on the vector $\mathbf{B_{PLS}}$ and as consequence in the vector w. This is done imposing an $L1$ term in this w vector in the optimization objective similar to the LASSO model. This means the method selects the most relevant predictors in the regression, making easier the understanding of the model, allowing also to perform a variable selection including only the variables with a regression coefficient different from zero [30] .

MLP-ANN Method. Artificial Neural Networks [41], and in particular the Multilayer Perceptron (MLP), are used to model non-linear systems. This is due to its ability to approximate any continuous function as accurately as necessary. Artificial neural networks are well-known algorithms applied in many fields with a large variety of architectures [42]. The trained weights that connect each input with the output through various neurons are used for the feature selection task [31,43]. So, for the MLP regression model of a variable x_i only the variables that have obtained a greatest score are take into account, i.e., the variables x_j with a score $R_{ij} > lnn_i$ have to be in the same block as x_i with lnn_i a certain threshold. This score is calculated as the product of the synaptic weights that connect each input with the output through the neurons in the neural model, i.e.,

$$R_{ij} = \sum_{k=1}^{H} W_{jk} W_{ki} \tag{7}$$

where R_{ij} is the relative importance, or score, of the input variable x_j, $j = 1, \dots, m$ and $j \neq i$ with respect to the output neuron, i.e., for the variable x_i that we are modeling, H is the number of neurons in the hidden layer, W_{jk} is the synaptic connection weight between the input neuron j and the hidden neuron k, and W_{ki} is the synaptic weight between the hidden neuron k and the output neuron.

SPCA method. The Sparse Principal component analysis (SPCA) method is also based on PCA, and, given a data matrix $\mathbf{X}$ and its covariance $\mathbf{S_x}$, the objective of this method is to decompose $\mathbf{S_x}$ into its components $[\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_d}]$ while constraining the number of elements of each vector $\mathbf{v}$ different from zero to r, so r is the cardinality of the vector $\mathbf{v}$. This is calculated maximizing the variance of $\mathbf{v} \in \mathfrak{R}^d$ with the constraint of the cardinality, i.e.,

$$\begin{aligned} &maximize\ \mathbf{v}^{\mathbf{T}}\mathbf{S_x}\mathbf{v} \\ &subject\ to\ \|\mathbf{v}\|_2 = 1 \\ &\qquad\qquad Card(\mathbf{v}) \leq r \end{aligned} \tag{8}$$

In general, in PCA technique only the first b components explain the most of the data variance. Here, SPCA is applied to the matrix $\mathbf{X}$ to obtain B sparse components, i.e., $[\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_B}]$ defining each component a block, and the non-zero coefficients of each sparse component, $\mathbf{v_i}, i = 1, \dots, B$ define the variables for that block [19].

Correlation. The relationship between two variables x_i and x_j can be calculated as the absolute value of the correlation coefficient :

$$\mathfrak{R}_i(x_i, x_j) = \left| x_i^T x_j \right| / \|x_i\| \|x_j\| \tag{9}$$

where $i, j = 1, \dots, m$. This coefficient is a direct measure of the correlation between variables and, after the calculation of vector $\mathfrak{R}_\mathbf{i}$ for the ith variable, the features selection for defining a block is carried out by selecting the variables with higher correlation values, i.e., the variables x_j with $\mathfrak{R}_\mathbf{i} > \delta_i$, are the selected variables to include for the block associate to the variable x_i, with δ_i being a cut off parameter for every variable that depends on the correlation present in the system [23].

Mutual Information. The mutual information of two variables measures their dependence considering the entropy [44]. In contrast to cross-correlation, it takes into account the higher-order statistics and is able to capture the non-Gaussianity of stochastic systems. So, this quantification would not only consider the linear correlations, but also takes into account the non-linear relations between variables. So, this information can be used to decompose the plant, i.e., variables with high mutual information values should be chosen to form the corresponding block [22]. MI can be calculated as:

$$I(x_1,x_2)=\int\int_{x_1x_2} p(x_1,x_2)\log(\frac{p(x_1,x_2)}{p(x_1)p(x_2)})dx_1dx_2 \tag{10}$$

where $p(x_1,x_2)$ is the joint probability density function while $p(x_1)$ and $p(x_2)$ are the marginal probability density functions of x_1 and x_2, respectively.

The features selection for defining a block, in this case, is carried out by selecting the variables with higher mutual information values, i.e., the variables x_j with $I(x_i,x_j) > Imi_{ij}$, are the selected variables to include for the block associate to the variable x_i, with Imi_{ij} being a threshold to be defined.

Minimal redundancy maximal relevance. Maximal relevance maximizes the relevance of the variables in the set S with respect to the target c as [45]:

$$max D(S,c) => D=\frac{1}{|S|}\sum_{x_i\in S} I(x_i,c) \tag{11}$$

where $|S|$ represents the number of variables in S and $I(x_i,c)$ is the mutual information between x_i and c. Note that the selected variables with maximal relevance could have rich redundancy. Therefore, the minimal redundancy rule must be added to select mutually exclusive variables,

$$min R(S) => R=\frac{1}{|S|^2}\sum_{x_i,x_j\in S} I(x_i,x_j) \tag{12}$$

So, in order to optimize D and R simultaneously, the function $\Phi(D,R)=D-R$ is defined. Now, the variables that maximized $\Phi(D,R)$ (Eq. (13)) would be in the same block as the target variable c.

$$max\Phi(D,R) => \max_{x_j\in X-S_{m-1}}\left[I(x_j,c)-\frac{1}{m-1}\sum_{x_j\in S_{m-1}} I(x_i,x_j)\right] \tag{13}$$

Here, the minimal redundancy maximal relevance (mRMR) algorithm not only considers the relevance but also reduces the redundancy among the abundant variables in plant-wide processes [28,45].

DCCA method. Detrended Cross-Correlation Analysis (DCCA) is a modification of the standard covariance analysis in which global average is replaced by local trends [46,47]. The DCCA coefficient for two variables x and y is calculated as the relation between the covariance without trend between x and y (F^2_{xy}) and the covariance without trend of x and y, i.e.,

$$\rho_{DDCA(x,y)}=\frac{F^2_{xy}}{F_xF_y} \tag{14}$$

The range of this coefficient is [−1 1], taking a value of 0 when there is no correlation between the variables. Here, the DCCA coefficient is calculated for each two variables in the process, and the variables with a large value for this coefficient are grouped together in a block.

Clustering. Clustering is an unsupervised machine learning approach for detecting groups of elements (clusters) according to some type of similarity or nearness, i.e., the main objective is to find clusters that minimize the inter-cluster variability and maximize the intra-cluster variability. The cluster algorithm used in this work is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [48], that is a density-based clustering algorithm, which uses the hyper-parameter ϵ for checking the neighborhood, or density of points, around each point, thus permitting clusters of arbitrary shape to be discovered.

Table 1 offers a brief comparison among all the discussed data-driven methods to divide the plant variables into blocks, briefly described in this section, in which the basic nature of the method (linear o nor lineal), the parameters to be defined and the technique used are described.

2.2. Canonical Variate Analysis

Canonical Variate Analysis (CVA) is a well-known multivariate method that maximizes the correlation between two sets of variables. It has been proposed for multivariate statistical analysis and was also developed for identifying state-space models [49]. Consider time series output data $\boldsymbol{y}(t)\in\Re^{m_y}$ and input data $\boldsymbol{u}(t)\in\Re^{m_u}$, in the instant $t\in(1,\dots,n)$, the past vector $\boldsymbol{p}(t)$ containing past outputs and inputs is defined as:

$$\boldsymbol{p}(t)=[\boldsymbol{y}^T(t-1),\boldsymbol{y}^T(t-2),\dots,\boldsymbol{y}^T(t-l),\boldsymbol{u}^T(t-1),\boldsymbol{u}^T(t-2),\dots,\boldsymbol{u}^T(t-l)]^T \tag{15}$$

while the future vector $\boldsymbol{f}(t)$, comprising the outputs in the present and future, is:

$$\boldsymbol{f}(t)=[\boldsymbol{y}^T(t),\boldsymbol{y}^T(t+1),\dots,\boldsymbol{y}^T(t+h)]^T \tag{16}$$

For an assumed state order k, the CVA algorithm computes an optimal matrix $\boldsymbol{J}_k$ that linearly relates the past vector $\boldsymbol{p}(t)$ to the reduced state vector $\boldsymbol{x}_k(t)\epsilon\Re^k$, via the singular value decomposition as:

$$\left(\boldsymbol{\Sigma}_{pp}\right)^{-1/2}\boldsymbol{\Sigma}_{pf}\left(\boldsymbol{\Sigma}_{ff}\right)^{-1/2}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}} \tag{17}$$

where $\boldsymbol{\Sigma}_{pp}$, $\boldsymbol{\Sigma}_{ff}$ and $\boldsymbol{\Sigma}_{pf}$ are the covariances of $\boldsymbol{p}(t)$, $\boldsymbol{f}(t)$ and the cross-covariance of $\boldsymbol{p}(t)$ and $\boldsymbol{f}(t)$, respectively. $\boldsymbol{\Sigma}$ is the diagonal matrix of non-negative singular values with descending order, $\boldsymbol{U}$ and $\boldsymbol{V}$ are matrices of the right and left singular vectors, and so the matrix $\boldsymbol{J}_k$ is obtained by

$$\boldsymbol{J}_{\boldsymbol{k}}=\boldsymbol{U}^{\boldsymbol{T}}_{\boldsymbol{k}}\left(\boldsymbol{\Sigma}_{pp}\right)^{-1/2} \tag{18}$$

where $\boldsymbol{U}_k$ contains the first k columns of $\boldsymbol{U}$, so the state vector $\boldsymbol{x}_k(t)$ is:

$$\boldsymbol{x}_k(t)=\boldsymbol{J}_k\boldsymbol{p}(t)=\boldsymbol{U}^T_k(\hat{\boldsymbol{\Sigma}}_{pp})^{-1/2}\boldsymbol{p}(t) \tag{19}$$

The values for l and h, i.e., the lags to include in the input vectors, and the state order k is not known a priory, however they can be calculated by the lags and the order that minimize the Akaike information criterion [50].

In the CVA method, two types of statistics are used to detect the behavior of the system: T^2_s, which measures the variations in the canonical subspace, and T^2_r, for the variations inside the residual subspace [8,51]:

$$\begin{aligned}T^2_s&=\boldsymbol{x}^T_k(t)\boldsymbol{x}_k(t)\\T^2_r&=\boldsymbol{x}^T_r(t)\boldsymbol{x}_r(t)\end{aligned} \tag{20}$$

where $\boldsymbol{x}_r(t)=\boldsymbol{J}_r\boldsymbol{p}(t)=\boldsymbol{U}^T_r(\hat{\boldsymbol{\Sigma}}_{pp})^{-1/2}\boldsymbol{p}(t)$ and $\boldsymbol{U}_r$ are the remaining $l(m_u+m_y)-k$ columns of $\boldsymbol{U}$ after extracting $\boldsymbol{U}_k$. The state of the process is determined using the thresholds of these statistics [8]. Another possibility to detect faults is using the residual vector:

$$\boldsymbol{r(t)}=(\boldsymbol{I}-\boldsymbol{J}^T_k\boldsymbol{J}_k)\boldsymbol{p}(t) \tag{21}$$

which allows the statistic Q to be obtained:

$$Q(t)=\boldsymbol{r}^{\boldsymbol{T}}\boldsymbol{r} \tag{22}$$

2.3. Bayesian Inference (BI)

In this decentralized CVA method, each block returns its own fault indexes, i.e., the three statistics defined in Eqs. (20)–(22) for each block. Then, it is necessary to fuse these multiple monitoring outcomes to obtain a global outperformed result. Various decision fusion strategies

Table 1
Characteristics of the data-driven process decomposition methods.

Method	Linear/Non-linear	Parameters	Technique
LASSO	Linear	λ	Regression
Elastic net	Linear	λ, δ, len_i: thresholds	Regression
SPLS	Linear	λ_1: sparsity	Regression
MLP	Non-linear	H: number of neurons in the hidden layer, lnn_i: thresholds	Regression
SPCA	Linear	r: cardinality, LV: number of latent variables	Information theory
Correlation (C1 and C2)	Linear	δ_i, lco_{ij} : thresholds	Information theory
Mutual Information (MI1 and MI2)	Non-linear	lmi_{ij}: thresholds	Information theory
mRMR	Non-linear	$lrmrm_{ci}$: thresholds	Information theory
DCCA (DCCA1 and DCCA2)	Linear	$ldcca1_i$, $ldcca2_{ij}$: thresholds	Information theory
DBSCAN	Non-linear	ϵ	Clustering

can be used, though the Bayesian Inference (BI) is the most popular one to fuse fault indexes [23,24,52,53], giving a single result for the whole plant.

In this fusion strategy the fault probability of the statistic ST (ST can be T_s^2, T_r^2 or Q) in each block i ($i = 1, 2, \dots, B$), B being the number of blocks, can be calculated by (where N denotes the normal condition, and F denotes the faulty condition)

$$P_{ST}(F \mid x_i) = \frac{P_{ST}(x_i \mid F)P_{ST}(F)}{P_{ST}(x_i)} \tag{23}$$

where

$$P_{ST}(x_i) = P_{ST}(x_i \mid N)P_{ST}(N) + P_{ST}(x_i \mid F)P_{ST}(F) \tag{24}$$

$P_{ST}(N)$ and $P_{ST}(F)$ are the prior probabilities of the process being normal and faulty, which can be simply assigned with confidence level α and $1-\alpha$ respectively [24,52,53]. The conditional probabilities $P_{ST}(x_i \mid N)$ and $P_{ST}(x_i \mid F)$ are calculated as:

$$P_{ST}(x_i \mid N) = e^{(-ST_i/ST_{i,lim})}, P_{ST}(x_i \mid F) = e^{(-ST_{i,lim}/ST_i)} \tag{25}$$

where ST_i represents the statistic ST of the current sample in the ith block and $ST_{i,lim}$ is the corresponding threshold for the statistic ST in the block i.

The BII index for ST in the whole plant is obtained by fusing all local results:

$$BII_{ST} = \sum_{i=1}^{B} \left\{ \frac{P_{ST}(x_i \mid F)P_{ST}(F \mid x_i)}{\sum_{i=1}^{B} P_{ST}(x_i \mid F)} \right\} \tag{26}$$

If the BII value for the statistic ST is over $(1-\alpha)$, a fault is detected with this statistic.

3. Comparative study

In this section a very summarized description is carried out about the decentralized monitoring scheme for testing the alternative decomposition approaches. This is made up of three steps: data driven process decomposition into blocks, local fault detection based on the CVA method for each block, and decision making based on all local results provided. This overall decentralized monitoring framework is show in Fig. 1 and the implementation is explained in the following subsection. The performance comparison is carried out over the Tennessee Eastman process (TEP). The TEP model is a well-known realistic model of a chemical plant and benchmark for control and monitoring studies [54].

3.1. Decentralized monitoring scheme

(A) Off-line procedure.

- *Step 1.*- Collect the training data under normal operation conditions, scale it to zero mean and unit variance, and construct the input matrix $\boldsymbol{X}$.
- *Step 2.*- Decompose the input data matrix $\boldsymbol{X}$ into B blocks using one of the methods defined in Section 2: such as LASSO regression, Elastic net regression, SPLS regression, ANN-MLP, SPCA, Correlation, Mutual Information, Maximal relevance minimal redundancy (mRMR), Detrended Cross-Correlation Analysis or DBSCAN clustering.
- *Step 3.*- Use a local monitoring scheme in each block. In this case, a CVA dynamic monitoring method is carried out, taking into account the auto and cross-correlated data of the industrial plants. Calculate the threshold for each statistical index and for each block.

(B) On-line Global monitoring procedure.

- *Step 1.*- Collect test data from the plant, normalize the data and build the input test matrix $\boldsymbol{X_t}$.
- *Step 2.*- Divide the matrix $\boldsymbol{X_t}$ into B blocks by the decomposition method chosen in the off-line procedure.
- *Step 3.*- Apply the CVA model in each block, calculating the T_{ri}^2, T_{si}^2 and Q_i statistics in each block $i = 1, \dots, B$. Compare the statistics with their corresponding thresholds.
- *Step 4.*- Combine the monitoring results of each block using the Bayesian inference index to obtain global $BII_{T_r^2}$, $BII_{T_s^2}$ and BII_Q, which are used to detect faults in the whole plant.
- *Step 5.*- If some of the BII indexes overpass the threshold $(1-\alpha)$, a fault is detected.

3.2. Indexes for comparison the process decomposition methods

- *False Alarms Rate (FAR)* takes into account the robustness of each statistic. It is the percentage of non-faulty samples classified as faulty. It can be calculated as:

$$FAR = 100\frac{N_{N,F}}{N_N}\% \tag{27}$$

  where $N_{N,F}$ is the faultless samples identified as faults, and N_N is the number of faultless samples.
- *Missed Detection Rate (MDR)*, quantifying the sensitivity to possible faults. It denotes what percentage of faulty measures are classified as faultless samples, and can be calculated as:

$$MDR = 100\frac{N_{F,N}}{N_F}\% \tag{28}$$

  where $N_{F,N}$ is the number of fault samples identified as normal, and N_F is the number of fault samples.
- *Fault Detection Delay (FDD)*. This measures how many samples are needed to detect a fault after its occurrence.
- *Number of Faults Detected (NFD)*.
- *Number of blocks (NB)* of the corresponding decomposition method.

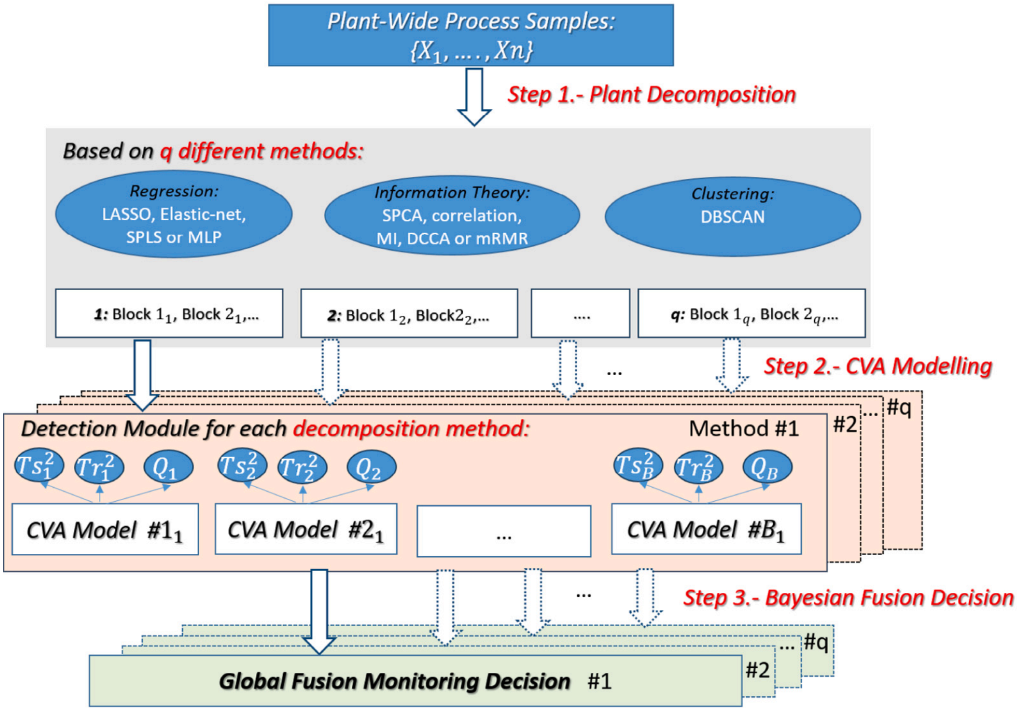
Plant-Wide Process Samples: $\{X_1, \ldots, Xn\}$
Step 1.- Plant Decomposition
Based on q different methods:
Regression: LASSO, Elastic-net, SPLS or MLP
Information Theory: SPCA, correlation, MI, DCCA or mRMR
Clustering: DBSCAN
1: Block 1_1, Block 2_1,...
2: Block 1_2, Block2_2,...
....
q: Block 1_q, Block 2_q,...
...
Step 2.- CVA Modelling
Detection Module for each decomposition method:
Method #1
#2
...#q
Ts_1^2 Tr_1^2 Q_1
Ts_2^2 Tr_2^2 Q_2
Ts_B^2 Tr_B^2 Q_B
CVA Model $\#1_1$
CVA Model $\#2_1$
...
CVA Model $\#B_1$
...
Step 3.- Bayesian Fusion Decision
Global Fusion Monitoring Decision #1
#2
...#q

Fig. 1. The CVA-based decentralized monitoring framework.

3.3. *Tennessee Eastman process benchmark*

The Tennessee Eastman Process is a well-known benchmark for monitoring and control [54,55]. Fig. 2 shows the flow diagram of the process with five major units, i.e., reactor, condenser, compressor, separator and stripper.

The process has two products from four reactants; an inert product and a by-product are also present, making, a total of 8 components denoted as A, B, C, D, E, F, G and H. The process allows 52 measurement, of which 41 are process variables and 11 are manipulated variables [55]. The data sets given in [50] are widely accepted for process monitoring studies. These available data, which can be downloaded from http://web.mit.edu/braatzgroup, are formed by 22 training sets (including normal and 21 fault operation conditions) collected to record the process measurements for 24 operation hours; and the corresponding 22 test data sets including, 48 h of the operation plant, in which the faults were generated after 8 simulation hours. The cited 21 faults are included in Table 2.

4. Results & Analysis

4.1. *Experimental setup*

First, the methods involved in this comparison (see Section 3, A-Step 2) have been tuned using the standard experimental methodologies applicable for each one, such as a cross validation and minimizing the root-mean-square-error (rMSE). Also, the thresholds defined for each method ($len_i, lnn_i, lmi_{i,j}, etc.$) are calculated in a grid search looking for the best results in terms of fault detection, i.e., the best results in terms of FAR, MDR and FDD for each method.

Notice that, as a CVA method is used, a state space model is calculated for every block to monitor the process; so, in this case, the variables chosen to perform every block must be the least correlated variables. On the other hand, the CVA hyper-parameters, l and h, the lags in the past and future vectors and the k canonical variables (CV) to be retained to form the state x_k (Eq. (19)) for each block have been tuned to maximize fault success and minimize fault dismissal. The local thresholds used for the statistics for each block were adjusted to obtain a significance level of 99%.

Finally, the threshold for the global decision index (BII index) and the number of consecutive alarms (N) necessary to detect a fault were adjusted for each method through a grid search for the best performance in terms of fault detection, and particularly N is tuned to obtain zero real false alarm rate. The threshold for all the methods is $\alpha = 0.9$, and N is defined for every method as explained below.

- **LASSO** [31]. A model is carried out for every system variable. Its hyperparameter λ from Eq. (2) is adjusted for every case to minimize $rMSE$ using a 3-cross validation procedure. Then, a block is generated for every variable x_i ($i = 1, \ldots, m$): including its own variable and those others with a zero coefficient in the respective LASSO regression model (i.e., the least relevant variables). The CVA parameters were $l = 4$ and $h = 4$ for most cases, and $N = 5$ consecutive anomalous observations are necessary to detect a fault.
- **Elastic net (EN)** [31]: An Elastic net based model was tuned for every variable of the system, using 3-cross validation and looking for the minimum average rMSE. The parameters δ and λ of Eq. (3) were individually tuned for every model in a grid search of both parameters. The block for a certain variable was made up of that variable and those others with coefficients from its respective model below a certain threshold, len_i, calculated as the mean value of the coefficients in each model. The order for the CVA models in each block, i.e., the parameters l and h for Eqs. (15) and (16) were such that 2, and $N = 6$ consecutive anomalous observations are necessary to detect a fault.
- **SPLS** [30]. An SPLS based model for each system variable is generated, taking as output $Y = x_i$ ($i = 1 \ldots m$) and X without x_i as predictor. Different values of sparsity were checked, i.e., how many coefficients in B_{iPLS} are considered zero, in order to minimize $rMSE$ using a 3-cross validation procedure. A block is created for each x_i ($i = 1, \ldots, m$) including its own variable and those others with zero coefficient in its corresponding SPLS model. Here, the CVA parameters were $l = 4$ and $h = 4$, and $N = 5$ consecutive anomalous observations are used to detect a fault.

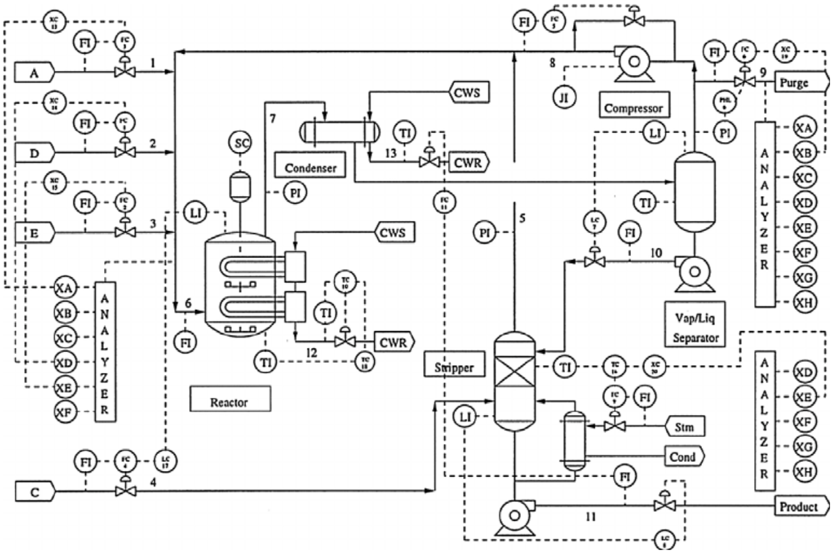

Fig. 2. Tennessee Eastman process diagram.

Table 2
Tennessee Eastman process faults.

Fault	Description	Type
1	A/C feed ratio, B composition constant (Stream 4)	Step
2	B composition, A/C ratio constant (Stream 4)	Step
3	D feed (Stream 2)	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss (Stream 1)	Step
7	C header pressure loss-reduced availability (Stream 4)	Step
8	A, B and C compositions (Stream 4)	Random variation
9	D feed temperature (Stream 2)	Random variation
10	C feed temperature (Stream 4)	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16	Unknown	–
17	Unknown	–
18	Unknown	–
19	Unknown	–
20	Unknown	–
21	Stream 4 valve	Sticking

- **MLP** [31]. A model for each system variable is built by an MLP-ANN. The MLP hyperparameters were considered in a wide range, as well as a 3-cross validation procedure to minimize $rMSE$. The block for a variable is made up of that variable and those that have obtain the smallest scores [31] by the model, i.e., the variables with a score R_{ij} see (Eq. (7)) below a certain threshold, lnn_i . This threshold was calculated as the mean value of the relevance values of the variables in each model. Here, the CVA order for each block was $l = 5$ and $h = 5$, and $N = 5$ consecutive anomalous observations are used to detect a fault.
- **SPCA**[19]. An SPCA model is built using all the system variables. The cardinality of the non-zero eigenvector elements (r, see Eq. (8)) was selected as $r = 40$, using a 3-cross validation procedure to obtain the best model. The obtained r eigenvectors were ranked and taking the first $B = 26$ latent variables that explain 70% of the variability of the faultless behavior, defining the resulting plant blocks. Zero indexes in every preserved eigenvector define each block structure, i.e., the variables considered in that block. The CVA parameters in each block were $l = 3$ and $h = 3$, and $N = 3$ consecutive alarms are used to detect a fault.
- **Correlation**:
  - **C1** [23]. Each block i ($i = 1, \dots, m$), being m the number of variables, includes all the variables whose correlation value with variable i does not overpass a certain threshold, i.e., such that $\mathbf{\Re_i}(x_j, x_i) < \delta_i \quad j = 1, \dots, m \quad j \neq i$ with $\delta_i = a \cdot median(\mathbf{\Re_i})$. Here, $a = 0.3$ to control the group size and choose it in a trial and error procedure to obtain the best results in the fault detection performance and $\mathbf{\Re_i}$ is the ith row of the correlation matrix. The CVA parameters are $l = 3$ and $h = 3$, and $N = 3$ consecutive alarms were necessary to detect a fault.
  - **C2**. In this case, the correlation $\Re$ is calculated, and the variables with minimal correlation regarding x_i go to the x_i block, i.e., if $\mathbf{\Re}_i(x_i, \mathbf{x_j}) < lco_{i,j}$, where $lco_{i,j} = 0.3 * median(\mathbf{\Re_i})$, then x_j is in the same block as x_i, giving disjoint blocks and fewer blocks than variables. The CVA parameters for each block are $l = 3$ and $h = 3$, and $N = 3$ consecutive alarms are used to detect a fault.
- **Mutual Information**.
  - **MI1** [22], MI_{x_i,x_j} being the mutual information between two variables x_i and x_j. If $MI_{x_i,x_j} <= lmi_{ij}$, with ($i =$

$1, \ldots, m; j = 1, \ldots, m$) then x_i and x_j are in the same block. lmi_{ij} is empirically tuned in a grid search to obtain the best results from the fault detection performance, resulting a value of $lmi_{ij} = 1.3 * I_{im}$ where I_{im} the median value of $MI_{x_i,x_j}, j = 1, \ldots, m$. The order for the CVA models are $l = 3$ and $h = 3$ and $N = 3$ consecutive alarms are used to detect a fault.

  - **MI2**. In this case, a block is obtained for each variable i, including all the variables whose mutual information values regarding i does not overpass a certain threshold, i.e., each block i ($i = 1, \ldots, m$) includes the variables x_j such that $MI_{x_i,x_j} < 0.05$. This threshold is tuned by a trial and error procedure to obtain the best results in the fault detection performance. If a block is made up of fewer than 4 variables, then this is removed and an extra block is created containing all these variables. Similar to the MI1 case, the CVA parameters are $l = 3$ and $h = 3$, and $N = 3$ consecutive alarms are used to detect a fault.

- **Maximal relevance minimal redundancy (mRMR)**[28]. Selecting arbitrarily a variable in X as a target variable c, for example the first variable x_1; if x_i, ($i = 2, \ldots, m$) satisfies $\Phi > lmrmr_{c,i}$ then x_i must be in the same block as c. If k variables are selected to be in this block, the rest of the variables are $X - X_k$, with X_k being the set with the k selected variables. Now, the variable with the minimal Φ in the remain dataset ($X - X_k$) is selected as the new target variable c, and the procedure is repeated until all the variables are in a block. The threshold $lmrmr_{c,i}$ is selected experimentally looking for the best result in the fault detection procedure, where $lmrmr_{c,i} = I_{mean}$ and I_{mean} is the mean value of mRMR through the variables with the same target. The CVA parameters in each block are $l = 5$ and $h = 5$, and $N = 6$ consecutive alarms are used to detect a fault.
- **Detrended Cross-Correlation**:
  - **DCCA1**: the ith row of the DCCA matrix is the DCCA coefficient of x_i regarding all other system variables. So, the ith block is composed of x_i and the variables in the ith row lower than the threshold $ldcca1 = 0.75 * median(\rho_{\mathbf{DDCA_i}})$, tuned by a trial and error procedure to obtain the best results in terms of the fault detection performance, i.e., minimum FAR and MDR. Here, CVA models are featured by $l = 3$ and $h = 3$, and $N = 3$ consecutive alarms are necessary to detect a fault.
  - **DCCA2**, with fewer blocks than variables. The variables x_j with $\rho_{\mathbf{DDCA_i}}(\mathbf{x_i}, \mathbf{x_j}) < ldcca2_{i,j}$ are in the same block as x_i, where $ldcca2_{i,j} = 0.25 * median(\rho_{\mathbf{DDCA_i}})$. The CVA models are parametrized by $l = 5$ and $h = 5$, and $N = 3$ consecutive alarms are needed to detect a fault.

- **DBSCAN** [32]. The kurtosis and skewness of each variable, the mean and variance of the correlation, and the mutual information of the variables are used by DBSCAN based clustering with standard validation indexes. The hyperparameter neighbor ratio (ϵ) is tuned to $\epsilon = 0.2$. The CVA models are parametrized by $l = 5$ and $h = 5$, and $N = 3$ consecutive alarms are used to detect a fault.

4.2. Results

Here the main results of the proposed comparison are shown and analyzed: a decentralized monitoring, previously dividing the plant into blocks, and a CVA-based fault detection method are implemented for each block. Then, all the outcomes are fused by Bayesian Inference Index. Simultaneously, the different decentralized methods are compared regarding the centralized CVA method proposed by [8]. The indexes for comparison described in Section 3 are used here to check the performance of every method. The results shown in the following subsections have been obtained in this work, except the results of the centralized CVA method which are shown as in [8], where the hyperparameters of the CVA method have been tuned to obtain the best performance.

4.2.1. Block decomposition: number of blocks (NB)

This is a major challenge for most of the decentralized monitoring approaches. Table 3 shows the number of blocks implemented by each method. The decentralized methods based on regression (LASSO, Elastic Net (EN), SPLS, neural networks (MLP)), Correlation (C1), Mutual Information (MI2) and DCCA1 work with one block per variable, so they all generate 52 blocks in this case study. This can be a serious issue when the system has so many variables.

On the other hand, other methods have fewer blocks than variables, such as SPCA (26 blocks), Mutual Information MI1 (10), correlation C2 (5), Maximal relevance minimal redundancy mRMR (10) and DCCA2 (4). So, these techniques are able to reduce the number of blocks generated. Finally, the DBSCAN clustering method can deliver a decentralized approach with a reduced number of blocks as small as 3.

So, if the number of blocks is an important issue in the application, it is best to choose a decomposition method with fewer blocks than variables, as MI1 or mRMR. Otherwise, if there were enough computing resources the method with best results should be chosen.

4.2.2. Fault detection and Fault Alarm Rate (FAR)

Table 3 shows that all the decentralized proposals detect more faults than the centralized CVA method, particularly LASSO regression, SPCA, C1 and MI1, which detect all 21 faults by the T_s^2 statistic. DBSCAN was also capable of detecting 20 faults by the T_r^2 and Q statistics, as central CVA only gets to detect 18 faults in the best case.

The fault alarm rates (FAR) in % calculated as Eq. (27) for the test data are also shown in Table 3, taking values from 0 to 4.8 for all the methods, clearly much lower than the central CVA for all the statistics. Notice that methods such as LASSO regression, SPLS, Neural networks (MLP), SPCA and DCCA2 show $FAR = 0$ for almost all the statistics.

4.2.3. Missed Detection Rate (MDR)

Tables 4–6 show the results of the Missed Detection Rate (MDR) for all the methods, including the centralized CVA method, for statistics $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q, respectively.

These tables show three groups of MDR results according to the faults.

- **Statistics** $BII_{T_s^2}$, Table 4:
  - **Faults 1–2, 4–8, 12–14 and 17–18**: all the methods offer very low MDR, except the centralized CVA for faults 4 and 7 and SPCA and Correlation (C2) for fault 5. So, these faults are very easy to detect, because the effects of the faults in the measured variables are high, and all the methods have good results.
  - **Faults 10–11, 16, 19–21**: the methods based on LASSO regression and MLP outperform the fault detection rate of the remaining methods. However, both methods decompose the plant into 52 blocks (as many blocks as variables), which can be a serious issue in large-sized plants. Among the methods with a lower number of blocks, the best is the decentralized method based on maximal relevance and minimal redundancy (mRMR), followed by the method based on mutual information (MI1).
  - **Faults 3, 9 and 15**: all the methods provide high MDRs, so they can not detect the faults, but even for this case, the LASSO regression based method is the best.

Table 3
Number of blocks (NB), False Alarm Rate in % (FAR) and Number of Faults Detected (NFD).

	LASSO	EN	SPLS	MLP	SPCA	C1	C2	MI1	MI2	mRMR	DCCA1	DCCA2	DBSCAN	CVA [8]
NB	52	52	52	52	26	52	5	10	52	10	52	4	3	1
FAR T_s^2	0	4.79	0	0	0.2	2.7	0.94	1.36	4.69	2.8	0.2	0	1.67	8.3
FAR T_r^2	0	1.77	0	0	0	3	2.29	2.6	1.56	1.8	0	0	4.1	12.6
FAR Q	0.4	1.4	0.2	0	0	1.14	0.52	1.77	0.83	3.5	0.8	0	3.55	8.7
NFD T_s^2	**21**	19	19	19	**21**	**21**	19	**21**	20	18	19	19	19	18
NFD T_r^2	19	18	19	19	18	**20**	**20**	19	18	18	**20**	**20**	**20**	18
NFD Q	18	18	19	18	19	18	18	19	19	18	18	17	**20**	17

Table 4
Missed Detection Rate (MDR) - $BII_{T_s^2}$.

Fault	LASSO	EN	SPLS	MLP	SPCA	C1	C2	MI1	MI2	mRMR	DCCA1	DCCA2	DBSCAN	CVA [8]
1	0.4	0.4	0.4	0.2	0.4	0.2	0.4	0.2	0.4	0.5	0.5	0.4	0.4	0.1
2	1.5	1.6	1.4	1.7	2.1	1.6	1.6	1.4	1.6	1.7	1.7	1.9	1.9	1.1
4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.4	4.4	5.4	68.8
5	0.1	0.1	0.1	0.1	67.1	4.4	74.1	0.1	0.1	0.1	0.1	0.25	0.1	0
6	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0
7	0.1	0.1	0.1	0.1	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.1	0.1	38.6
8	2.1	1.8	2.5	1.6	2.2	2.1	2.4	2.6	2.5	2.1	2.2	2.4	2.5	2.1
12	0.2	0.2	0.2	0.2	0.6	0.4	0.7	0.2	0.2	0.4	0.4	0.4	0.2	0
13	4.4	4.6	4.5	4	4.9	4.6	4.9	4.6	4.9	5.1	4.5	5.5	5.1	4.7
14	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0
17	2.7	3.1	3.2	2.4	3.2	3	3.7	2.6	5.7	2.7	3	6.7	9.7	10.4
18	8.4	9.2	9.4	9.2	9.4	9.6	9.9	10	9.4	9.9	9.4	9.7	9.7	9.4
10	7.2	8.2	10	6.7	15.2	9.7	11	13	15.6	6.4	9.6	20.1	10.6	16.6
11	12.4	22.2	18.9	10.4	12.5	15.4	24.6	20.1	24.7	16.1	22.2	35.4	35.5	51.5
16	2.7	8.6	4.6	3.6	8.25	7.2	8.4	26.2	10.5	4.5	3.7	19.9	9.7	16.6
19	2.2	10.1	11.2	0.5	9.9	2.5	37.4	13.2	40.5	15.7	42.1	55.2	55.5	84.9
20	9.5	10.2	12.3	8.5	20.25	20	31.1	14.1	17.5	12.1	17.1	25.2	22.4	24.8
21	42.5	41.9	48.1	30.2	56.6	57	56	50.2	51.2	46.1	48.1	53.6	57.5	44
3	89.2	94.2	97.2	96.5	95	96.2	98.7	98.6	94.4	97.9	92.7	99.5	97.1	98.1
9	92.9	95.6	97	98.4	95.9	98.1	9.1	98.5	95.9	98	94.5	99.7	99.1	98.6
15	75.7	84.2	88.2	78.9	91.1	94.6	97.5	93.4	89.6	96.6	84.9	98.2	90.2	92.8
Mean	**16.9**	18.9	19.5	**16.8**	23.6	20.3	22.5	21.5	22	19.8	20.8	25.6	24.3	31.6
MMDR*	**5.4**	6.9	7.1	**4.4**	11.8	7.6	14.8	9	10.1	6.9	9.2	13.2	12.6	20.7

MMDR* = Mean MDR for 18 faults (without faults 3, 9 and 15).

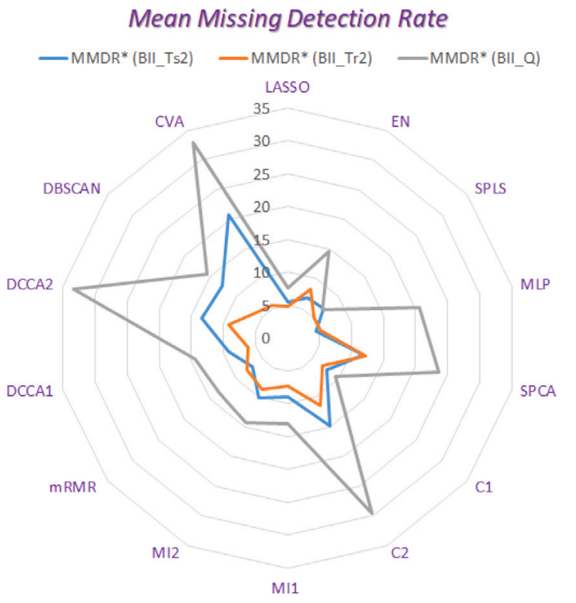

Fig. 3. MDR (MMDR*) average for the $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q statistics in each method.

For a more intuitive comparison, the average missed alarm rate is shown in Fig. 3 (the graph in blue) and in Table 4, specifically in the last two rows of the table, for all the faults (21) and (MMDR*) for 18 faults (excluding faults 3, 9 and 15). Considering $BII_{T_s^2}$, the methods based on LASSO (5.4) and MLP-ANN (4.4) regression obtain clearly better results, followed by the Elastic net regression (6.9), mRMR (6.9), SPLS (7.1) and correlation (C1 with a value of 7.6) based methods.

On the other hand, the centralized CVA method [8] is better for some faults (5, 6, 12 and 14) with $MDR = 0$ (although very similar to other methods such as LASSO, SPLS, MLP-ANN, Elastic net, mRMR or MI1 and MI2 with values $MDR = 0.1$ or $MDR = 0.2$). However considering the mean of all the faults, the centralized CVA method is the worst with a large difference.

The best MMDR*s, considering methods with a lower number of blocks are for the mRMR with a value of 6.9 followed by the Mutual Information (MI1, with a value of 9) and SPCA (11.8) methods.

- **Statistics $BII_{T_r^2}$**, Table 5:
  - **Faults 1–2, 4–8, 12–14 and 17–18**: all the methods offer very low MDR, except for the SPCA and C2 methods in fault 5 that have a very high value. Also the value for the mRMR method in the same fault is high, 15.1, compared with the rest of the methods. Note that the best results are for the centralized CVA method, close to $MDR = 0$, but this MDR is very similar to the remaining methods (MDR = 0.1 or MDR = 0.2). This result is because the FAR for CVA is 8.12 [8], a very high value with regard to the other methods (see Table 3), i.e., this statistic threshold is very low, and a very high FAR and a very low MDR values are obtained.

Table 5
Missed Detection Rate (MDR) - $BII_{T_r^2}$.

Fault	LASSO	EN	SPLS	MLP	SPCA	C1	C2	MI1	MI2	mRMR	DCCA1	DCCA2	DBSCAN	CVA [8]
1	0.2	0.2	0.2	0.2	0.4	0.2	0.4	0.2	0.4	0.5	0.2	0.4	0.2	0
2	1.1	1.7	1.2	1.4	2	1.9	1.7	1.5	1.9	1.4	2	1.6	1.6	1
4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0
5	0.1	0.1	0.1	0.1	71.2	0.2	70.2	0.1	0.1	15.1	0.1	0.2	0.1	0
6	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0
7	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0
8	1.7	2.5	1.9	1.7	2.2	2	2.1	2.4	2	2.7	2.4	2.5	2.6	1.6
12	0.2	0.2	0.2	0.2	0.9	0.2	0.4	0.2	0.2	1.1	0.2	0.4	0.2	0
13	4.4	4.5	4.4	4.2	4.9	4.5	4.7	4.5	5	5.1	4.6	4.7	4.7	4
14	0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0
17	2.5	2.7	2.5	2.4	3.7	2.6	2.4	2.5	5.1	3	2.5	2.9	3.5	2.4
18	9.4	9.7	9.5	9.4	9.7	9.2	9.2	9.4	9.7	9.4	9.4	9.4	9	9.2
10	7	13.4	8.2	7.5	18.1	10.1	9.9	17.9	15.5	10.2	9.6	57.7	7.6	9.9
11	10.7	18.7	12.2	10.6	10.9	10.2	13.6	8.1	19.2	5.2	7.6	11.5	11.5	19.5
16	3.6	11	5.1	4.5	10.5	5.7	6.9	17.5	14	6.6	3.2	8.5	5.2	8.4
19	0.6	4.1	1.6	0.4	1.6	0.6	1.9	1.2	8.6	5.9	1.1	0.7	1.1	1.9
20	8.6	18.6	9.1	8.5	22.8	18	22.2	11.1	16	18.4	10.5	14.2	8.2	8.7
21	35.1	58.6	36.6	40.7	59.7	55.7	59	53.6	58.6	57.2	55.2	49.2	54.5	34.2
3	97.5	98.4	97.6	96.9	98.5	95.6	95.4	96.1	98.1	98.5	97.7	98	93	98.6
9	98.9	99	98.6	98.5	98.4	96.7	96.1	98.4	99.1	98.9	96.7	98.7	96	99.3
15	82	93	86	82.1	98.1	95.5	95.9	88.5	85.2	99.5	94.9	98.1	79.2	90.3
Mean	**17.3**	20.82	**17.9**	**17.6**	24.5	19.5	23.5	19.7	21.41	20.95	18.9	21.9	18	18.5
MMDR*	**4.8**	8.2	**5.2**	**5.1**	12.2	6.8	11.42	7.2	8.7	7.9	6.1	9.1	6.1	5.6

MMDR* = Mean MDR for 18 faults (without faults 3, 9 and 15).

  - **Faults 10–11, 16, 19–21**: the results are very similar for all the methods but the best are those based on regression (LASSO, SPLS and MLP) and also those based on DBSCAN and Detrended Cros-correlation Analysis (DCCA1). However, the CVA centralized method is not very behind.
  - **Faults 3, 9 and 15**: the MDRs are very high and, really, these faults are not detected by any of the methods.

  Observing the average missing alarm rate (MMDR*) for all the faults in Table 5 and Fig. 3 (the graph in red), the LASSO regression based method is the best one (4.8), followed by the methods based on MLP-ANN (5.1), SPLS (5.2) and centralized CVA (5.6). Taking into account the number of blocks, the best is the CVA centralized method, which has only one block, followed by the DBSCAN based decentralized method (6.1 and 3 blocks) and the mutual information based decentralized method (MI1, 7.2 but 10 blocks).

- **Statistics** BII_Q, Table 6:
  - **Faults 1–2, 4–8, 12–14 and 17–18**: low MDR values for almost all cases except the centralized CVA for faults 4, 7 and 8; MLP-ANN, EN and DBSCAN for fault 4; SPCA for faults 4 and 5; Correlation (C2) for faults 4, 5 and 7; and Detrended cross-correlation (DCCA2) for nearly all these faults. In this case, in general, the results with this statistic are not so good as with $BII_{T_r^2}$ and $BII_{T_s^2}$, as it is possible to see in Figure Fig. 3 (the graph in grey), clearly higher than the other statistics.
  - **Faults 10–11, 16, 19–21**: the methods based on SPLS and LASSO regression provide a better fault detection performance followed by the method based on correlation (C1). The method based on maximal relevance minimal redundancy also gives good results for these faults, except for faults 19 and 21, which are not detected by this method due to the high MDR obtained for these faults. Notice that MLP-ANN does not give as good results as for the other statistics, maybe because MLP-ANN obtains a non-linear model for each variable, extracting non-linear characteristics of the plant to build the blocks. $BII_{T_s^2}$ monitors the behavior of the model and BII_Q monitors the residual space, so if the model is better, then it is expected that $BII_{T_s^2}$ will be more effective than BII_Q.
  - **Faults 3, 9 and 15**: MDR is very high, i.e., these faults are not really detected by any of the methods, but even in this case, the methods based on linear regression, i.e., LASSO and SPLS, obtain better results.

  Observing MMDR* for all the faults in Table 6 and Fig. 3 (the graph in grey), it is possible to see that the methods based on linear regression (SPLS and LASSO) are the best with values 6.8 and 7.5 respectively followed by the method based on correlation (C1, with MDDR* = 9.4). If the number of blocks was an index to be taken into account, the best results would be for the method based on mutual information (MI1), followed by the mRMR method, which has a slightly different value, i.e., 13.1 and 13.2 respectively, followed by the method based on DBSCAN (15.5).

Comparing the MDR results for the correlation methods (C1 and C2) regarding the three statistics, it is possible to note significantly better results always for the method C1, i.e., considering a block per variable as in [23]. Nevertheless, if the MI1 and MI2 methods are considered, the best results are always for the MI1 method, i.e., the method with fewer blocks than variables (10 blocks), as proposed by [22] but the differences in this case are not so much significant. It is also possible to note that the DCCA method shows slightly better results than the correlation based methods for statistics $BII_{T_r^2}$ and $BII_{T_s^2}$, but they are worse for the BII_Q statistic, so no relevant improvement is achieved by this complex method, at least for this plant.

Finally, if MI1 method is compared with the mRMR method, which is an improvement over the mutual information method [27] , the results are very similar. In the $BII_{T_s^2}$ statistics, the mRMR method wins with a median value of 6.9 compared to MI1, that has a value of 9, but for the other two statistics, $BII_{T_r^2}$ and BII_Q, the values are similar (7.2 and 13.1 respectively for the MI1 method and 7.9 and 13.2 for the mRMR method), so in this example nearly any improvement has been obtained.

4.2.4. Fault Detection Delay (FDD)

The detection delay is required to be as low as possible, so the monitoring scheme has to detect the faults as soon as possible. From

Table 6
Missed Detection Rate (MDR) - BII_Q.

Fault	LASSO	EN	SPLS	MLP	SPCA	C1	C2	MI1	MI2	mRMR	DCCA1	DCCA2	DBSCAN	CVA [8]
1	0.2	0.6	0.4	0.5	1	0.4	1.1	0.5	0.4	1.1	0.5	1.1	0.7	0.3
2	1.6	2.9	1.5	2.5	3.1	2.1	3.2	2	2	2.4	2.4	5.5	1.5	2.6
4	0.1	39.4	0.1	84.8	46.2	7.7	20.9	3.6	17.7	0.1	24	98.9	23.5	97.5
5	0.1	0.1	0.1	0.1	76.4	0.4	83.5	0.1	0.1	0.5	0.1	0.5	0.1	0
6	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0
7	0.1	0.1	0.1	10.4	0.1	0.1	74	0.1	0.1	0.1	0.1	19.6	0.1	48.6
8	2.2	3.6	2	3.4	5.4	2.6	11.9	2.9	2.4	7.2	2.5	20.7	2.2	48.6
12	0.2	0.2	0.2	0.2	5.9	0.6	15.4	0.4	0.2	0.6	0.4	6.9	0.4	2.1
13	4.9	5.4	4.2	4.1	8.9	4.7	8.5	5.4	5.1	7.7	6.2	16.5	5.5	5.5
14	0.2	0.3	0.2	2.7	0.6	0.2	1	0.2	0.4	0.1	0.4	4.2	0.2	12.2
17	3.6	8.7	4	3.9	19.2	5.1	11	6.7	10.1	5	6.6	25.9	12.5	13.8
18	9.2	10.2	8.4	9.5	10.5	9.7	11.1	9.9	9.7	10.7	9.5	10.9	9.5	10.2
10	10	17.1	9.6	16.9	38.5	8.4	21.6	33.1	10.9	9.1	17.2	49.9	24.1	59.9
11	17.5	47.4	15.1	45.1	50.1	27.6	40	36.6	35.4	7.1	41	77.4	25.4	66.4
16	6.5	15.5	4.7	18.4	27.2	8	20.2	34.6	10.2	8.4	12	52.9	20.9	42.9
19	10.1	31.2	4.9	62.5	35	3	80.1	20.4	52.1	68.2	41.9	85.7	62.9	92.3
20	16.2	15.5	16.4	34.8	24.4	21.5	43.2	17.2	40.1	20.1	31.9	39.2	31	35.4
21	51.9	64.9	50.6	71.4	71.5	67	85.9	61.9	62.1	83.4	61.2	85.9	58.8	54.8
3	93.2	97.4	90.5	97.9	99.1	98.7	99.1	95.9	99	97.9	98.5	99	93.8	98.5
9	93.1	98	88.7	98.9	99.4	99	98.5	98.1	99.4	98.5	98.5	99	90.9	99.3
15	89.1	97.7	85.4	98.5	96.6	98.2	99.6	94.5	83.5	98	98.4	98.2	87.2	97.9
Mean	19.5	26.5	18.4	31.7	34.2	22.2	39.5	25	25.7	25.4	26.4	42.8	26.3	42.3
MMDR*	**7.5**	14.6	**6.8**	20.6	23.6	9.4	29.6	13.1	14.4	13.2	14.3	33.4	15.5	32.9

MMDR* = Mean MDR for 18 faults (without faults 3, 9 and 15).

an engineering point of view, a fault is usually indicated only when a determined number of consecutive values of the statistics exceed their thresholds [8]. In this work, this number of consecutive values, N, for each of the methods is described in Section 4.1, and the detection delay is recorded at the first time instance for which the control limit is exceeded. Tables 7–9 and Fig. 4 show the detection delays for statistics $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q, respectively.

- **Statistics** $BII_{T_s^2}$, Table 7:
  Table 7 is also organized in three fault sets. Here however, every method shows similar detection delay, except the centralized CVA method for faults 4 and 11, which shows very high values regarding the remaining methods, and for fault 19, which is not detected by CVA.
  To look for the best method, we use the average detection delay (MFDD*) for 18 faults excluding, faults 3, 9 and 15 (last row of Table 7) and Fig. 4. Considering $BII_{T_s^2}$, the MLP-ANN based regression is the best one, (26,5) with a slightly difference with regard to the LASSO based regression (with value 30) and the DCCA2 based method (30,2). Also, the centralized CVA method [8] obtains the worst results, showing a large MFDD* with regard to the rest of methods, even considering that fault 19 is not detected; so only 17 faults are included in calculation for MFDD* in this method (i.e., excluding faults 5, 9, 15 and 19). Considering the MFDD* for the methods with few blocks, the Detrended Cross-correlation Analysis (DCCA2) is the best followed by the method based on mutual information (MI1), which is slightly worse.

- **Statistics** $BII_{T_r^2}$, Table 8:
  The Fault Detection Delay of each method, including the centralized CVA method, is very similar for every fault. The only difference is the detection time in fault 21, where Mutual Information (MI1) is clearly the best for that fault, followed by DCCA2. The worst case is for the centralized CVA method.
  All this can be much clearly observed using the average value, MFDD* and Fig. 4: the best is the MI1 method, but no very significant differences are observed, except for the DBSCAN, C2, MI2, mRMR, EN, and centralized CVA methods, which are the worst.

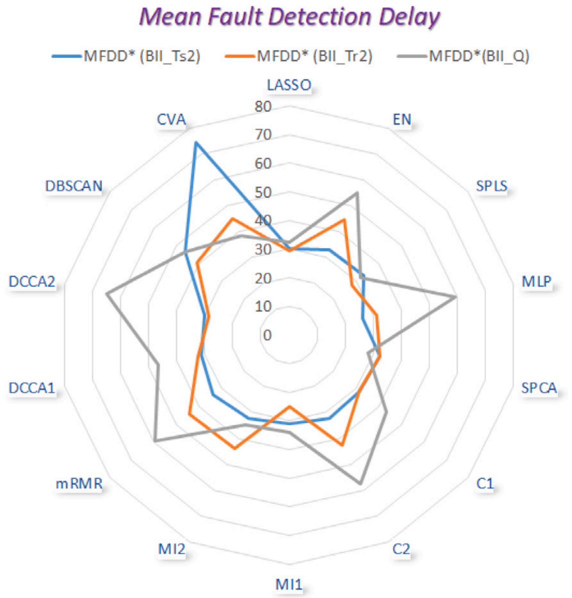

Fig. 4. FDD (MFDD*) average for the $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q statistics in each method.

- **Statistics** BII_Q, Table 9:
  The results are very similar for all the methods, with a few exceptions such as for fault 4 by EN and MLP-ANN with a very large FDD, or DCCA2 method for fault 11 and the non-detection of fault 4. The best MFDD*, see Fig. 4 and the last row of Table 9, is for the SPCA method (28.1), followed by the methods based on linear regression, i.e., SPLS (32.1) and LASSO (32.5). Note that the CVA method only takes 17 faults for MFDD* (i.e., excluding faults, 3, 9, 15 and 19), so this is not comparable with regard to the other methods.

In any case, taking into account the eight indexes considered (MDR and FDD for $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q), the false alarms rate (FAR) and the number of detected faults, the decentralized strategies achieved better results for almost all of these indexes than the centralized CVA

Table 7
Fault Detection Delay (FDD) - $BII_{T_s^2}$.

Fault	LASSO	EN	SPLS	MLP	SPCA	C1	C2	MI1	MI2	mRMR	DCCA1	DCCA2	DBSCAN	CVA [8]
1	3	4	3	2	3	2	2	2	3	5	4	3	3	2
2	13	14	13	9	15	15	13	11	13	16	15	15	13	13
4	1	2	1	1	1	1	1	1	1	2	1	1	1	461
5	1	2	1	1	1	1	1	1	1	2	1	2	1	1
6	1	2	1	1	1	1	1	1	1	2	1	1	1	1
7	1	2	1	1	1	1	1	1	1	2	1	1	1	1
8	18	17	20	17	19	17	21	18	20	18	19	19	20	20
12	2	3	2	2	3	3	3	2	2	4	3	3	2	2
13	36	38	37	32	39	37	39	37	39	42	39	42	44	42
14	1	2	2	1	1	1	1	1	1	2	1	2	2	2
17	19	20	20	19	23	20	21	20	23	23	21	21	22	27
18	77	80	78	76	76	79	80	80	81	81	79	78	79	83
10	22	24	23	22	21	21	22	22	21	23	21	23	22	25
11	7	12	7	7	7	7	7	7	7	7	7	8	7	292
16	9	10	9	9	10	9	9	14	11	10	9	14	10	14
19	3	12	12	3	11	2	12	11	11	12	11	15	47	
20	71	71	71	67	66	65	74	67	75	68	76	69	71	82
21	255	276	300	207	285	286	271	265	272	286	255	227	490	275
3	43				45	51		45	51		41			
9	5				6	731		5						
15	577	676	635	3	241	241	681	631	679		575	706	639	677
MFDD*	**30**	32.8	33.4	**26.5**	32.4	31.5	32.2	31	32.3	33.7	31.3	30.2	46.5	79

MFDD* = Mean value of fault detection delay for 18 faults (without faults 3, 9 and 15).

Table 8
Fault Detection Delay (FDD) - $BII_{T_r^2}$.

Fault	LASSO	EN	SPLS	MLP	SPCA	C1	C2	MI1	MI2	mRMR	DCCA1	DCCA2	DBSCAN	CVA [8]
1	2	3	2	2	2	2	3	2	3	5	2	3	2	3
2	11	15	11	12	16	15	15	12	15	12	16	13	13	15
4	1	2	1	1	1	1	1	1	1	2	1	1	1	1
5	1	2	1	1	2	2	1	1	1	3	1	2	1	1
6	1	2	1	1	1	1	1	1	1	2	1	1	1	1
7	1	2	1	1	1	1	1	1	1	2	1	1	1	1
8	17	21	18	18	20	16	19	16	19	23	19	20	19	20
12	2	3	2	2	3	2	2	2	2	3	2	3	2	2
13	35	39	35	34	39	36	39	36	41	45	38	38	39	39
14	1	2	1	2	1	1	1	1	1	2	1	2	1	1
17	20	20	20	19	23	20	20	20	23	21	20	21	20	20
18	76	80	77	76	80	76	79	76	79	77	80	76	76	79
10	22	24	22	22	25	21	24	24	24	24	23	36	20	23
11	7	12	7	7	6	6	6	6	7	8	7	7	6	11
16	9	9	9	10	9	8	9	11	10	10	8	10	9	9
19	2	12	3	3	2	2	2	2	11	13	11	2	2	11
20	66	66	67	66	65	65	65	65	72	66	67	66	65	66
21	257	490	226	287	286	286	484	174	479	474	286	211	456	511
3							484				54	586		
9													731	
15	643		643	3			741	391			746	132	659	
MFDD*	**29.5**	44.7	**28**	31.3	32.3	31.2	42.9	**25**	43.9	44.1	32.4	**28.5**	40.8	45.2

MFDD* = Mean of DD for 18 faults (without faults 3, 9 and 15).

method. So, all this shows that the decentralized approaches are preferable alternatives to centralized ones. In this comparison, the only index with good results for the centralized CVA method is the MDR with the T_r^2 statistic. However, if the division of the plant is not adequate the results are not so good, such as for the DCCA2 method, which is the worst decentralized method.

Comparing the decentralized methods, the LASSO regression based method is the best, followed by the SPLS based method, i.e., in general, the methods based on linear regression obtain the best results. So, these methods are the best to carry out the division of the plant, followed by the MLP-ANN regression based method with a very similar performance regarding $BII_{T_s^2}$ and $BII_{T_r^2}$, but not for BII_Q and the C1 based method. However, all these methods need a total decentralization of the plant, meaning the same number of blocks as variables. If the number of variables is very high, or it is necessary to reduce the number of blocks due to computational issues, the Mutual Information (MI1) is the best method with a lower number of blocks than variables.

5. Conclusions

In this paper, a comparative study of decentralized monitoring schemes based on CVA has been carried out. These decentralized methods divide the plant-wide process variables into several blocks, then statistical data models are built to perform local monitoring. All the local monitoring from every block is integrated through a decision fusion algorithm. In this work, different approaches for the plant variable decomposition are studied: some based on regression, such as LASSO, Elastic net, SPLS and MLP-ANN; others are based on information such as Sparse Principal Component Analysis (SPCA), Correlation, Mutual Information, Maximal Relevance and Minimal Redundancy (mRMR) and Detrended Cross-correlation Analysis (DCCA); and finally others are based on clustering such as DBSCAN method. Then, a Canonical Variate Analysis (CVA) based local dynamic monitor is set up for each of these established blocks, and finally the Bayesian Inference (BI) is introduced to achieve a global decision of fault detection for the whole

Table 9
Fault Detection Delay (FDD) - BII_Q.

Fault	LASSO	EN	SPLS	ANN	SPCA	C1	C2	MI1	MI2	mRMR	DCCA1	DCCA2	DBSCAN	CVA [8]
1	2	6	3	4	8	3	10	4	3	10	4	10	6	2
2	14	31	14	19	25	19	27	15	19	22	19	46	13	25
4	1	79	1	249	1	1	5	1	1	2	11		1	0
5	1	2	1	1	4	3	7	1	1	2	1	4	1	0
6	1	2	1	1	1	1	1	1	1	2	1	1	1	0
7	1	2	1	1	1	1	1	1	1	2	1	3	1	0
8	18	23	20	21	26	22	26	22	20	26	23	30	22	21
12	2	3	2	2	10	4	27	2	2	6	3	4	2	0
13	41	48	35	35	47	39	47	45	43	49	50	58	49	43
14	2	2	2	10	3	2	3	2	3	2	3	2	2	1
17	21	24	20	21	28	22	24	22	25	24	23	30	28	23
18	80	80	78	79	85	79	89	82	80	87	83	88	82	84
10	24	25	22	25	29	22	25	27	22	23	26	36	25	44
11	8	12	6	9	9	6	7	9	11	8	7	99	11	27
16	10	10	9	10	14	9	10	12	9	11	9	20	14	11
19	12	13	12	19	11	2	13	13	12	12	11	15	17	
20	71	65	68	74	64	64	76	65	77	68	76	78	73	72
21	276	561	283	489	140	480	639	288	299	712	492	648	485	302
3														
9			43									239		
15		676			118			677	46				524	
MFDD*	**32.5**	55.1	**32.1**	59.4	**28.1**	43.3	57.6	34	35	59.5	46.8	65.1	46.3	38.5

MFDD* = Mean FDD for 18 faults (without faults 3, 9 and 15).

process based on the information coming from the local monitors. All the discussed methods were tested over an industrial benchmark of TE process to complete a detailed comparison study. As far as the results are concerned, we would like to point out that:

- Taking into account all the indexes considered (MDR and Fault Detection Delay (FDD) for the three statistics $BII_{T_s^2}$, $BII_{T_r^2}$ and BII_Q, the False Alarms Rate (FAR) and the Number of Faults Detected (NFD)), the decentralized methods, in general, are better than the centralized CVA model. In this comparative, the only index with good results for the CVA method is the MDR with the T_r^2 statistic, but with a very high fault alarm rate, compared to the rest of the methods.
- The decomposition methods based on regression, in general, give lower MDRs in most of the faults than for the remaining methods in the comparative, due to their ability to capture the strong relationships between the variables. The methods based on linear regression, such as LASSO and SPLS, provide good results for all the indexes and all the statistics; while the method based on MLP-ANN (a non-linear regression method) provides low values for the statistics $BII_{T_s^2}$ and $BII_{T_r^2}$, but worse results for the BII_Q statistic, for both MDR and Detection Delay indexes. The method based on correlation (C1) also gives goods results, but they are slightly worse than the methods based on regression. However, all these methods decompose the plant into a very high number of blocks.
- If the number of blocks is an important index to take into account, the method based on Mutual Information (MI1) also gives low values for the MDR and detection delay indexes in all the statistics, and these results are better than the other method results, which have a lower number of blocks than variables, i.e., SPCA, C2, mRMR, DCCA2 and DBSCAN.
- Comparing the results and considering all the indexes for the correlation methods (C1 and C2), it is worth noting that the results are always better for the method C1, i.e., considering a block per variable as proposed by [23], instead of using fewer non-overlapping blocks with a clearly better perform. If a comparative between the Correlation methods and the Detrended Cross-Correlation analysis is also carried out, it can be see that the results are not very different, so no relevant improvement has been reached with this more complex analysis, at least for this case study.
- Comparing the results and considering all the indexes for the mutual information methods (MI1 and MI2), it is worth nothing that the results are always better for the method MI1, i.e., considering a lower number of blocks than variables, as proposed by [22] instead of using a block per variable as in the C1 method. If a comparison between the Mutual Information methods and the Maximal Relevance Minimal Redundancy (mRMR) is also carried out, it can be see that results between the methods MI1 and mRMR are not very different with the MDR indexes considering the three statistics; both divide the plant variables into 10 blocks, but the results for the fault detection delay (FDD) index and the three statistics are clearly better for the MI1, so no noticeable improvement has been obtained with this more complicated analysis, at least for this case study.

CRediT authorship contribution statement

M.J. Fuente: Conceptualization, Funding acquisition, Investigation, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review & editing. **M. Galende-Hernández:** Conceptualization, Investigation, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review & editing. **G.I. Sainz-Palmero:** Conceptualization, Funding acquisition, Investigation, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

None declared.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Spanish Government through the Ministerio de Ciencia e Innovación (MICINN), Spain / Agencia Estatal de Investigación (AEI), Spain under Grant (MICCIN/AEI/10.13039/501100011033) PID2019-105434RB-C32.

References

[1] S. Yin, S.X. Ding, X. Xie, H. Luo, A review on basic data-driven approaches for industrial process monitoring, IEEE Trans. Ind. Electron. 61 (11) (2014) 6418–6428, http://dx.doi.org/10.1109/TIE.2014.2301773.

[2] S. Yin, X. Li, H. Gao, O. Kaynak, Data-based techniques focused on modern industry: an overview, IEEE Trans. Ind. Electron. 62 (1) (2015) 657–667, http://dx.doi.org/10.1109/TIE.2014.2308133.

[3] Z. Ge, Review on data-driven modeling and monitoring for plant-wide industrial processes, Chemometr. Intell. Lab. Syst. 171 (2017) 16–25, http://dx.doi.org/10.1016/j.chemolab.2017.09.021.

[4] Q. Jiang, X. Yan, B. Huang, Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes, Ind. Eng. Chem. Res. 58 (2019) 12899–12912, http://dx.doi.org/10.1021/acs.iecr.9b02391.

[5] S. Yin, X. Zhu, O. Kaynak, Improved PLS focused on key-performance-indicator-related fault diagnosis, IEEE Trans. Ind. Electron. 62 (3) (2015) 1651–1858, http://dx.doi.org/10.1109/TIE.2014.2345331.

[6] Z. Luo, Y. Wang, S. Lu, P. Sun, Process monitoring using a novel robust PCA scheme, Ind. Eng. Chem. Res. 60 (11) (2021) 4297–4404, http://dx.doi.org/10.1021/acs.iecr.0c06038.

[7] G.L.P. Palla, A.K. Pani, Independent component analysis application for fault detection in process industries: Literature review and an application case study for fault detection in multiphase flow systems, Measurement 209 (2023) http://dx.doi.org/10.1016/j.measurement.2023.112504.

[8] E.L. Russell, L.H. Chiang, R.D. Braatz, Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis, Chemometr. Intell. Lab. Syst. 51 (1) (2000) 81–93, http://dx.doi.org/10.1016/S0169-7439(00)00058-7.

[9] Y. Zhang, Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (KICA) and support vector machine (SVM), Ind. Eng. Chem. Res. 47 (18) (2008) 6961–6971, http://dx.doi.org/10.1021/ie071496x.

[10] J. Huang, X. Yan, Dynamic process fault detection and diagnosis based on dynamic principal component analysis, dynamic independent component analysis and bayesian inference, Chemometr. Intell. Lab. Syst. 148 (2015) 115–127, http://dx.doi.org/10.1016/j.chemolab.2015.09.010.

[11] C. Chakour, M. Harkat, M. Djeghaba, New adaptive kernel principal component analysis for nonlinear dynamic process monitoring, Appl. Math. 9 (2015) 1833–1845.

[12] Y. Si, Y. Wang, D. Zhou, Key-performance-indicator-related process monitoring based on improved kernel partial least squares, IEEE Trans. Ind. Electron. 68 (3) (2021) 2626–2636, http://dx.doi.org/10.1109/TIE.2020.2972472.

[13] H. Wang, M. Peng, Y. Yu, H. Saeed, C. Hao, Y. Liu, Fault identification and diagnosis based on KPCA and similarity clustering for nuclear power plants, Ann. Nucl. Energy 150 (2021) 107786, http://dx.doi.org/10.1016/j.anucene.2020.107786.

[14] X. Sha, N. Diao, Robust kernel principal component analysis and its application in blockage detection at the turn of conveyor belt, Measurement 206 (2023) http://dx.doi.org/10.1016/j.measurement.2022.112283.

[15] Z. Ge, J. Chen, Plant-wide industrial process monitoring: a distributed modeling framework, IEEE Trans. Ind. Inform. 1 (2016) 310–321, http://dx.doi.org/10.1109/TII.2015.2509247.

[16] Y. Zhang, H. Zhou, S. Qin, T. Chai, Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares, IEEE Trans. Ind. Inform. 6 (1) (2010) 3–10, http://dx.doi.org/10.1109/TII.2009.2033181.

[17] Q. Liu, S.J. Qin, T. Chai, Multiblock concurrent PLS for decentralized monitoring of continuous annealing processes, IEEE Trans. Ind. Electron. 61 (11) (2014) 6429–6437, http://dx.doi.org/10.1109/TIE.2014.2303781.

[18] A. Sánchez-Fernández, M.J. Fuente, G.I. Sainz-Palmero, Fault detection in wastewater treatment plants using distributed PCA methods, in: 2015 IEEE 20th Conference on Emerging Technologies Factory Automation, ETFA, 2015, pp. 1–7, http://dx.doi.org/10.1109/ETFA.2015.7301504.

[19] M. Grbovic, W. Li, P. Xu, A. Usadi, L. Somg, S. Vucetic, Decentralized fault detection and diagnosis via sparse PCA based decomposition and maximum entropy decision fusion, J. Process Control 22 (2012) 738–750, http://dx.doi.org/10.1016/j.jprocont.2012.02.003.

[20] Z. Ge, Z. Song, Distributed PCA model for plant-wide process monitoring, Ind. Eng. Chem. Res. 52 (2013) 1947–1957, http://dx.doi.org/10.1021/ie301945s.

[21] C. Tong, X. Yan, A novel decentralized process monitoring scheme using a modified multiblock PCA algorithm, IEEE Trans. Autom. Sci. Eng. 14 (2) (2017) 1129–1138, http://dx.doi.org/10.1109/TASE.2015.2493564.

[22] Q. Jiang, X. Yan, Plant-wide process monitoring based on mutual information multiblock principal component analysis, ISA Trans. 53 (2014) 1516–1527, http://dx.doi.org/10.1016/j.isatra.2014.05.031.

[23] C. Tong, X. Shi, Decentralized monitoring of dynamic processes based on dynamic feature selection and informative fault pattern dissimilarity, IEEE Trans. Ind. Electron. 63 (6) (2016) 3804–3814, http://dx.doi.org/10.1109/TIE.2016.2530047.

[24] Y. Tian, T. Hu, X. Peng, W. Du, H. Yao, Decentralized monitoring for large-scale process using copula-correlation analysis and Bayesian inference based multiblock principal component analysis, J. Chemometr. 33 (8) (2019) 1–18, http://dx.doi.org/10.1002/cem.3158.

[25] Y. Tian, H. Yao, Z. Li, Plant-wide process monitoring by using weighted copula-correlation based multiblock principal component analysis approach and online-horizon Bayesian method, ISA Trans. 96 (2020) 24–36, http://dx.doi.org/10.1016/j.isatra.2019.06.002.

[26] M.-Q. Zhang, X. Jiang, Y. Xu, X.-L. Luo, Decentralized dynamic monitoring based on multi-block reorganized subspace integrated with Bayesian inference for plant-wide process, Chemometr. Intell. Lab. Syst. 193 (2019) 103832, http://dx.doi.org/10.1016/j.chemolab.2019.103832.

[27] C. Xu, S. Zhao, F. Liu, Distributed plant-wide process monitoring based on PCA with minimal redundancy maximal relevance, Chemometr. Intell. Lab. Syst. 159 (2017) 53–63, http://dx.doi.org/10.1016/j.chemolab.2017.08.004.

[28] K. Zhong, M. Han, T. Qiu, B. Han, Y.-W. Chen, Distributed dynamic process monitoring based on minimal redundancy maximum relevance variable selection and Bayesian inference, IEEE Trans. Control Syst. Technol. 28 (5) (2020) 2037–2044, http://dx.doi.org/10.1109/TCST.2019.2932682.

[29] B. Xiao, Y. Li, B. Sun, C. Yang, K. Huang, H. Zhu, Decentralized PCA modeling based on relevance and redundancy variable selection and its application to large-scale dynamic process monitoring, Process Saf. Environ. Prot. 151 (2021) 85–100, http://dx.doi.org/10.1016/j.psep.2021.04.043.

[30] A. Sánchez-Fernández, M.J. Fuente, G.I. Sainz-Palmero, Decentralized and dynamic fault detection using PCA and Bayesian Inference, in: 2018 IEEE 23th Conference on Emerging Technologies Factory Automation, ETFA, 2018, http://dx.doi.org/10.1109/ETFA.2018.8502656.

[31] M.J. Fuente, G. Sainz-Palmero, M. Galende-Hernández, Dynamic decentralized monitoring for large-scale processes using regression based multiblock canonical variate analysis, IEEE Acess 11 (2023) 26611–26623, http://dx.doi.org/10.1109/ACCESS.2023.3256719.

[32] A. Sánchez-Fernández, M.J. Fuente, G.I. Sainz-Palmero, Decentralized DPCA model for large-scale processes monitoring, in: 2019 IEEE 24th Conference on Emerging Technologies Factory Automation, ETFA, 2019, http://dx.doi.org/10.1109/ETFA.2019.8869128.

[33] J. Wang, P. Liu, S. Lu, M. Zhou, X. Chen, Decentralized plant-wide monitoring based on mutual information-Louvain decomposition and support vector data description diagnosis, ISA Trans. 133 (2023) 42–52, http://dx.doi.org/10.1016/j.isatra.2022.07.017.

[34] R. Paredes, T.J. Rato, L.O. Santos, M.S. Reis, Hierarchical statistical process monitoring based on a functional decomposition of the causal network, in: 32nd European Symposium on Computer Aided Process Engineering, ESCAPE32, Vol. 51, 2022, pp. 1417–1422, http://dx.doi.org/10.1016/B978-0-323-95879-0.50237-X.

[35] W. Yu, C. Zhao, B. Huang, MoniNet with concurrent analytics of temporal and spatial information for fault detection in industrial processes, IEEE Trans. Cybern. 52 (8) (2022) 8340–8351, http://dx.doi.org/10.1109/TCYB.2021.3050398.

[36] W. Yu, M. Wu, C. Lu, Meticulous process monitoring with multiscale convolutional feature extraction, J. Process Control 106 (2021) 20–28, http://dx.doi.org/10.1016/j.jprocont.2021.08.014.

[37] H. Seongmin, P. Daoutidis, Control-relevant decomposition of process networks via optimization-based hierarchical clustering, AIChE J. 62 (9) (2016) 3177–3188, http://dx.doi.org/10.1002/aic.15323.

[38] R. Tibshirani, Regression shrinkage and selection via the LASSO, B, J. R. Statist. Soc. 58 (1996) 267–288.

[39] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, B, J. R. Statist. Soc. 67 (2005) 301–320.

[40] H. Chun, S. Keles, Sparse partial least squares regression for simultaneous dimension reduction and variable selection, J. R. Stat. Soc. Ser. B Stat. Methodol. 72 (1) (2010) 3–25, http://dx.doi.org/10.1111/j.1467-9868.2009.00723.x.

[41] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer-Verlag New York, 2013.

[42] S. Haykin, Neural Networks and Learning Machines, Prentice Hall, 2009.

[43] J. de Oña, C. Garrido, Extracting the contribution of independent variables in neural network models: a new approach to handle instability, Neural Comput. Appl. 25 (2014) 859–869, http://dx.doi.org/10.1007/s00521-014-1573-5.

[44] M.M. Rashid, J. Yu, A new dissimilarity method integrating multidimensional mutual information and independent component analysis for non-Gaussian dynamic process monitoring, Chemom. Intell. Laborary Syst. 114 (2012) 44–58, http://dx.doi.org/10.1016/j.chemolab.2012.04.008.

[45] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238, http://dx.doi.org/10.1109/TPAMI.2005.159.

[46] B. Podobnik, H. Stanley, Detrended cross- correlation analysis: A new method for analyzing two nonstationary time series, Phys. Rev. Lett. 100 (8) (2008) 084102, http://dx.doi.org/10.1103/PhysRevLett.100.084102.

[47] G.B. Zebende, DCCA cross-correlation coefficient: quantifying level of cross-correlation, Physica A 100 (4) (2011) 614618, http://dx.doi.org/10.1016/j.physa.2010.10.022.

[48] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, 1996, pp. 226–231.

[49] W.E. Larimore, Statistical Methods in Control and Signal Processing, Marcel Dekker, 1997.

[50] L.H. Chiang, E.L. Russell, R.D. Braatz, Fault Detection and Diagnosis in Industrial Systems, Springer-Verlag London, 2001, pp. 103–112.

[51] B. Jiang, D. Huang, X. Zhu, F. Yang, R.D. Braatz, Canonical variate analysis-based contributions for fault identification, J. Process Control 26 (2015) 17–25, http://dx.doi.org/10.1016/j.jprocont.2014.12.001.

[52] C. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag New York, 2006.

[53] Z. Ge, M. Zhang, Z. Song, Nonlinear process monitoring based on linear subspace and Bayesian inference, J. Process Control 20 (5) (2010) 676–688, http://dx.doi.org/10.1016/j.jprocont.2010.03.003.

[54] A. Bathelt, N.L. Ricker, M. Jelal, Revision of the Tennessee Eastman Process model, in: 9th IFAc Symposium on Advanced Control on Chemical Process, ADCHEM2015, 2015, http://dx.doi.org/10.1016/j.ifacol.2015.08.199.

[55] J.J. Downs, E.F. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (1993) 245–255, http://dx.doi.org/10.1016/0098-1354(93)80018-I.