Digital Chemical Engineering 18 (2026) 100290

Contents lists available at ScienceDirect

Digital Chemical Engineering

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

Original article

All you need is noise — from feature selection to explainable industrial AI☆

Mattia Vallerio [a,*], Antonio del Rio Chanona [b], Francisco J. Navarro-Brull [b,c,**]

[a] Dipartimento di Chimica, Materiali e Ingegneria Chimica "Giulio Natta", Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, 20133, MI, Italy
[b] Imperial College London, London, SW7 2AZ, UK
[c] International Flavors & Fragrances Inc., IFF, Benicarló, 12580, Spain

ARTICLE INFO

Keywords:
Machine learning
Explainable AI
Synthetic noise features
Feature selection
Active learning
Bayesian optimization

ABSTRACT

Modern chemical plants record thousands of sensor tags, yet only a small fraction meaningfully influence yield, quality, or throughput. Identifying those key drivers is often more difficult than building the predictive model itself. In this work, we show that appending one or more *Synthetic Noise Features* (SNFs), non-informative random variables known *a priori*, provide a simple reference for judging variable relevance. We show the impact of this model agnostic step across three workflows. In supervised learning, noise features establish an automatic cutoff for the feature importance, guide model regularization and signal when the dataset itself lacks predictive information. In unsupervised learning, they provide an unbiased threshold preventing spurious anomalies and latent dimensions. Finally, we demonstrate the applicability of this approach to small datasets typical of experimental work and Design of Experiments (DoE), including *Definitive Screening*, *Response Surface*, and *space-filling* designs, as well as *active learning* using *Bayesian optimization*. By turning *nothing but noise* into a quantitative benchmark, SNFs offer an immediately deployable safeguard against overfitting and misplaced experimental effort in data-driven chemical engineering.

1. Introduction

1.1. Industrial context

Chemical engineering processes have many recipe parameters and sensors. This higher-dimensional feature space is true for all scales: from laboratory analysis up to the operation of an industrial-scale plant and the supply chain optimization of products and raw materials for a chemical production site. It is therefore not surprising that data analytics and machine learning methods have gained such high popularity in this research field, Qin (2014), Chiang et al. (2017), Mowbray et al. (2022), like many others. What we are experiencing is only the beginning, since the number of features, measured variables and/or mathematical reformulation of those variables, is only destined to grow given the improvements in sensor technologies, the introduction of wireless sensors (Lou et al., 2022), and the adoption of advanced sensors such as cameras (e.g., RGB, thermal, spectral), Stuyck and Demeester (2024). Additionally, for safety purposes, several features are measured with redundant sensors; this can be the case for temperature or pressure inside a reactor. Finally, the feature space of these applications is not only high-dimensional, but also complex due to the presence of features that can be highly non-linear, time and product-dependent, collinear or correlated, and noisy.

Data scale in practice. For example, a typical chemical manufacturing site may have between 20k and 100k sensors stored in data historians (time-series databases). During production, process engineers use these sensors, also referred to as tags, to monitor, troubleshoot, and optimize their manufacturing processes. While specialized software is starting to establish itself to facilitate quick analysis (e.g., Seeq, TrendMiner, PEPITe, etc.), this vast amount of industrial data imposes practical challenges. As the dimension of the feature space grows from tens to hundreds (or even thousands of sensors) of variables, irrelevant and redundant variables tend to proliferate.

These extraneous features can add noise, degrade model performance, and, more importantly, obscure critical process insights, Yin et al. (2014). In terms of data analytics, high-dimensional data will: (i) increase computational cost, (ii) heighten the risk of overfitting, and (iii) experience the *curse of dimensionality*, Bellman (1961), Guyon and Elisseeff (2003).

1.2. Why variable selection?

Variable selection (also referred to as variable elimination) is a well-known first step to address this issue. By identifying a smaller subset of

☆ This article is part of a Special issue entitled: 'Digital Process Industry' published in Digital Chemical Engineering.

* Corresponding author.
** Corresponding author at: Imperial College London, London, SW7 2AZ, UK.
E-mail addresses: mattia.vallerio@polimi.it (M. Vallerio), f.navarro@imperial.ac.uk (F.J. Navarro-Brull).

https://doi.org/10.1016/j.dche.2026.100290
Received 10 September 2025; Received in revised form 26 January 2026; Accepted 28 January 2026
Available online 4 February 2026

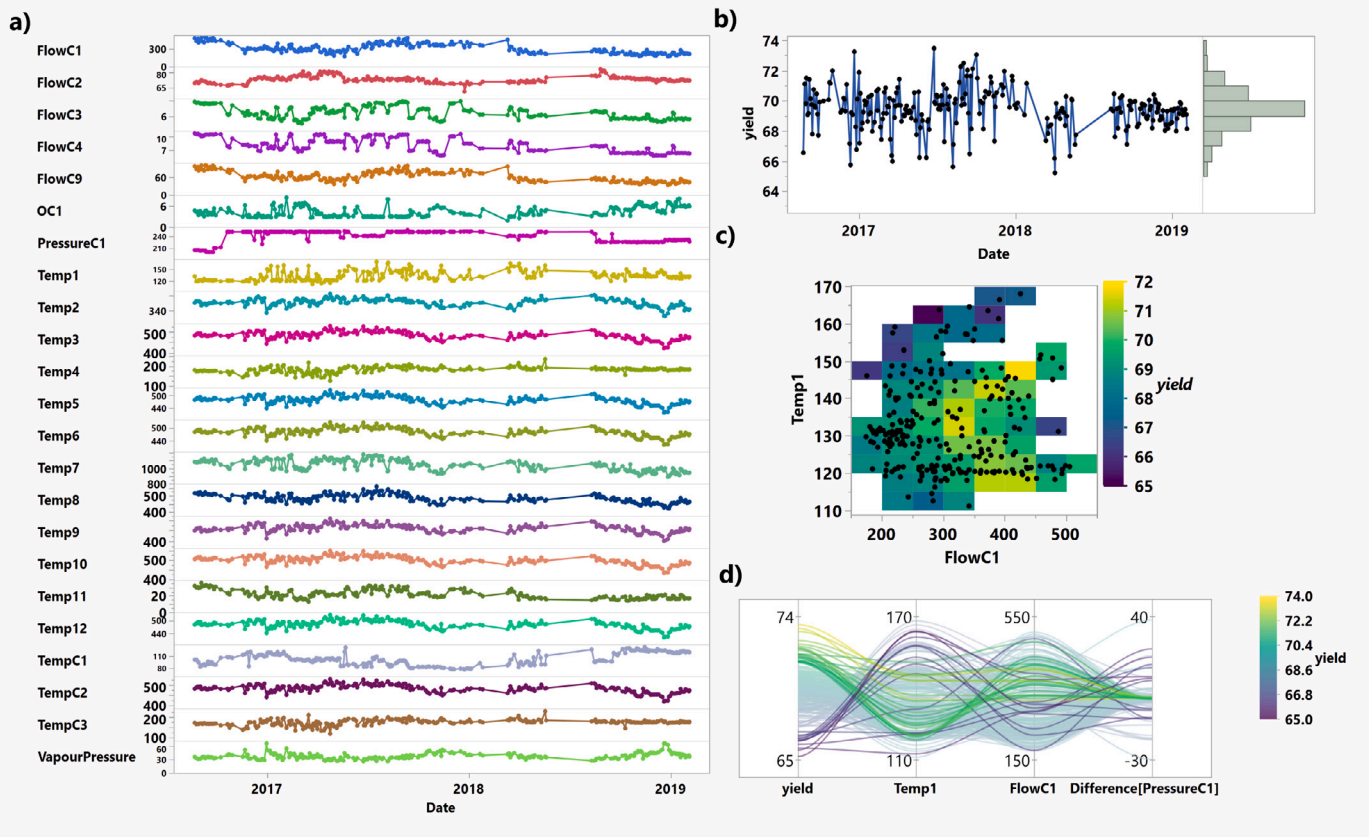

Fig. 1. Distillation-column data used for the supervised-learning examples: the target *yield* is a non-linear function of three measurements: **FlowC1**, **Temp1**, and the consecutive-sample pressure change Δ_tPressureC1 ≡ PressureC1(t) − PressureC1($t-1$). **(a)** Time trends for all instrumented tags across the sampled period; markers indicate timestamps where measurements are available, while the connecting line is shown only as a visual guide to the overall trend. **(b)** Calculated yield shown as a time series together with its empirical distribution (after scaling to 65%–74%); black markers indicate available measurements, while the blue line is shown only as a visual guide to the trend. **(c)** Heatmap of FlowC1 vs. Temp1 colored by yield, revealing the characteristic "banana-shaped" response induced by the Rosenbrock function. **(d)** Parallel coordinates plot for FlowC1, Temp1, and Δ_tPressureC1 (lines colored by yield): high yields cluster where $|\Delta_t\text{PressureC1}|$ is small, while low yields occur when the pressure change is large in either direction.

the most pertinent variables, a process expert can quickly identify potential causes. In contrast to dimensionality reduction techniques such as Principal Component Analysis (PCA) (Wold et al., 1987), reducing data dimensionality without creating any new transformed variables enhances interpretability and ensures the model's focus remains on key process drivers. This is achieved by eliminating irrelevant variables (e.g., instrument signals that offer no useful correlation or variation).

Synthetic Noise Features (SNFs): core idea. This paper builds on an established concept in the data science and machine learning community to introduce one or more Synthetic Noise Features (SNFs) and use them as a criterion for the feature selection procedure.

This simple but powerful concept is at the base of the well-known R package Boruta, Kursa and Rudnicki (2010). It has been widely used in multiple applications, including with datasets with hundreds or thousands of gene expression variables, Kursa (2014). In a similar vein, chemical process datasets, such as those derived from large-scale, continuous or batch reactors or multi-unit operations, can be equally complex, potentially housing many correlated or non-informative variables, Mowbray et al. (2022), Arzac-Garmendia et al. (2022), Arzac et al. (2023), Vallerio et al. (2024), Zuecco et al. (2021)

Related work. Closely related research strands include permutation-based feature importance (originating in Random Forests and extended to conditional and bias-corrected forms), Breiman (2001), Strobl et al. (2008), Altmann et al. (2010), synthetic *knockoff* random variables, see Stoppiglia et al. (2003), Barber and Candès (2015) enabling false-discovery rate controlled selection, Candès et al. (2018), Barber et al. (2020), Weinstein et al. (2017), permutation tests assessing end-to-end model significance, Ojala and Garriga (2010), and variable selection, Wu et al. (2007) that have been applied in linear models Barber and Candès (2019), Support Vector Machines (SVMs), boosted trees Jiang et al. (2020), and Bi et al. (2003), Artificial Neural Networks (ANN) Lu et al. (2018). These methods use synthetic or permuted variables as negative controls or reference baselines, typically within specific tasks (e.g., importance ranking or selective inference).

1.3. Scope and structure

Beyond variable selection, we extend this concept and demonstrate its applicability with supervised, unsupervised, and active learning problems. Additionally, we show practical examples that highlight the validity of the presented approach in the industrial context. The proposed SNFs method can be applied to all existing machine learning methods without the need to modify them.

The remainder of this work is structured as follows: Section 2 will introduce the proposed methodology and its application to supervised learning problems. Section 3 will extend the concept to unsupervised learning problems and show its usefulness. The application of the proposed methodology for active learning methods will be covered in Section 4. Finally, Section 5 will draw the conclusions and detail the possible future research lines extending from this work.

2. Supervised learning

In this section, we show the application of the proposed methodology of adding Synthetic Noise Features (SNFs) to supervised machine learning problems. For this purpose, we will first motivate the need for this methodology and will introduce some relevant use cases for the process industry.

2.1. Motivation

Key performance parameters such as yield, throughput, or cycle time are commonly monitored. In case a certain variation is detected, engineers troubleshoot and optimize using their process expertise and data-driven techniques. Very often, the first approach would be to see if the observed variability is correlated with either the control of manipulated variables or their response to disturbances. This correlation or regression analysis of input variables affecting and output (target) is what is known as supervised learning.

In practice, we rarely know the root cause in advance (Mowbray et al., 2022; Zuecco et al., 2021; Arzac et al., 2023). Modern plants are highly coupled — with recycle streams, heat-integration networks, inventory dynamics, and advanced process control (APC/MPC) layers — so small changes can ripple through units and constraints, obscuring the true driver. When a deviation appears, teams must select a manageable subset of candidate variables to investigate, which means some plausible causes are inevitably excluded (e.g., utilities are often assumed healthy and omitted). SNFs provide a practical safeguard in this setting. In this section, we show how to apply SNFs to supervised problems to (i) set an automatic cutoff for variable relevance and (ii) serve as a regularization/penalization signal during model training.

2.2. Supervised learning datasets

The application of the SNFs methodology for supervised learning is demonstrated on an open-source distillation-column dataset, Duhn (2025). The dataset shown in Fig. 1, contains 253 observations (data rows) collected over ~ 2.5 years with 23 columns with sensors (tags). We treat product *yield* as the target for regression and construct it as a simple non-linear function of a few key drivers,

$$\text{yield} = f(\text{FlowC1}, \text{Temp1}) - \left|\Delta_t \text{PressureC1}\right|,$$

where $f(\cdot)$ follows a Rosenbrock function of $\{\text{FlowC1}, \text{Temp1}\}$, a standard non-convex benchmark with a characteristic banana-shaped valley (see Fig. 1c). The resulting yield response is subsequently log-scaled and standardized with an offset to lie between 65% and 74%.

$\Delta_t \text{PressureC1} \equiv \text{PressureC1}(t) - \text{PressureC1}(t-1)$ denotes the consecutive-sample change in column pressure (not a spatial pressure drop ΔP). To emulate measurement variability and unmodeled effects, we add independent noise: Gaussian ($\mu = 0$, $\sigma = 0.3$) and uniform over $[-0.15, 0.15]$. The remaining columns are unaltered plant tags and serve as candidate predictors for feature screening and model fitting. This dataset and yield construction were previously used in Mowbray et al. (2022) and can be downloaded from our repository (Vallerio and Navarro-Brull, 2025).

2.3. Variable selection

A feature selection procedure typically comprises two steps. First, choose a selection of a criterion. This metric or heuristic (e.g., correlation with the output, mutual information, predictor performance) is used to assess the relevance of each feature. Second, search for a subset of relevant features, Blum and Langley (1997). Since evaluating all subsets of features becomes infeasible as the feature set grows, a suboptimal but computationally tractable search strategy (filter, wrapper, or embedded) is employed, Li et al. (2017). Filter methods rank features using statistical criteria before modeling, Guyon and Elisseeff (2003), wrapper methods use the predictor performance itself to evaluate different subsets, Saeys et al. (2007), and embedded methods integrate feature selection within the model-training process, Chandrashekar and Sahin (2014).

Adding a synthetic noise feature (SNF) to perform variable selection in supervised machine learning problems is an already established practice, Kursa and Rudnicki (2010). The idea here is that the SNFs will function as a reference to classify the other features based on their relative importance factor concerning the target variable. Therefore, this method removes the need to define a selection criterion, and it falls in the wrapper category, Barber and Candès (2015). Moreover, it can be used without modifying any algorithm. Fig. 2 shows the proposed SNFs method applied to three different algorithms: (i) Random/Bootstrap Forest (100 trees) (Breiman, 2001), (ii) Boosted Tree (Friedman, 2000), (iii) Partition Tree (Breiman et al., 2017). The features are ranked starting from the most important to the least one and are shown based on the rank obtained with the Bootstrap Forest algorithm. All three methods correctly identify Temp1 and FlowC1 as the most important features. The synthetic target variable is a non-linear Rosenbrock function of these two features, see Section 2.2. As can be seen, the added SNFs maintain the same importance order of magnitude for the three ML methods tested here. It will now be possible to perform a feature selection just by removing all variables that show a lower or equal importance than the first SNF in order of importance.

Excluding these variables from the analysis can help to:

1. **Minimize noise**: Excess features may introduce bias and reduce the ability of the model to generalize, Guyon and Elisseeff (2003).
2. **Reduce computational overhead**: Smaller subsets of variables make the training and implementation of models more efficient, an important consideration for real-time monitoring and control, Chandrashekar and Sahin (2014).
3. **Improve prediction accuracy**: When only the most relevant features are considered, models often provide more reliable forecasts of product quality, yield, and plant performance, Saeys et al. (2007), Bi et al. (2003).
4. **Enhance process insight**: Feature selection can highlight the underlying factors that most strongly influence performance, safety, or product quality, guiding engineers toward more targeted process improvements, Qin (2014).

Similarly, the SNFs can also give an indication if there is a sufficient amount of information in the dataset. In particular, if, during training, the SNFs rank highest in feature importance, it indicates the real inputs do not contain enough information to explain the target. This could be due to a lack of variation in the dataset or a limited number of data points available. To illustrate this concept, we used the same dataset as in Fig. 2, but this time our target variable was shuffled (same values of the original variable but in a random order). This permutation implies that the target variable is now a random variable itself, but it holds the same distribution as the original one. Fig. 3 shows the results of the screening model with the "shuffled" yield variable as the target.

2.4. Penalization

The SNFs could also be used as a regularization or penalization, a ML modeling practice that usually requires data partitioning (Mowbray et al., 2022). However, once an algorithm starts to assign importance to one of the added SNFs features, this is already a sign that the resulting model might be overfitting the information in the training dataset. Therefore, the parameters of the ML model should be reduced. This could be easily illustrated with a partition tree example, see Fig. 4. The first time the algorithm selects the SNF for a split (gray circle at 12 splits in Fig. 4) is taken as the *noise threshold*. This can be seen as a first indication of overfitting, since the algorithm is trying to use an

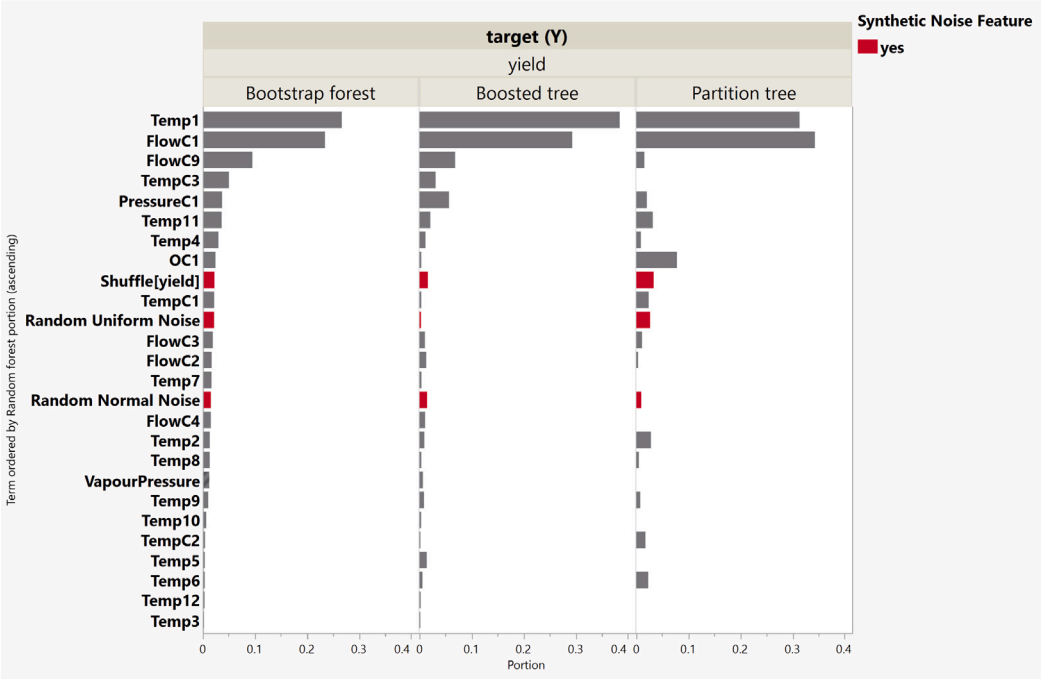

Fig. 2. Feature-importance bar chart for the distillation-column dataset (product yield as the target), where variables are ranked by their relative contribution to the total fitted variation explained by the model. The synthetic-noise feature (SNF, shown in red) provides a reference threshold that cleanly separates Temp1 and FlowC1 from all other tags, confirming that only these two process variables consistently outrank pure noise across models. Results are shown for three models: a Random/Bootstrap Forest with 100 trees ($R^2 = 0.81$), a Gradient-Boosted Tree ensemble with 144 trees and two splits per tree ($R^2 = 0.67$), and a single Partition Tree with 38 splits ($R^2 = 0.79$).

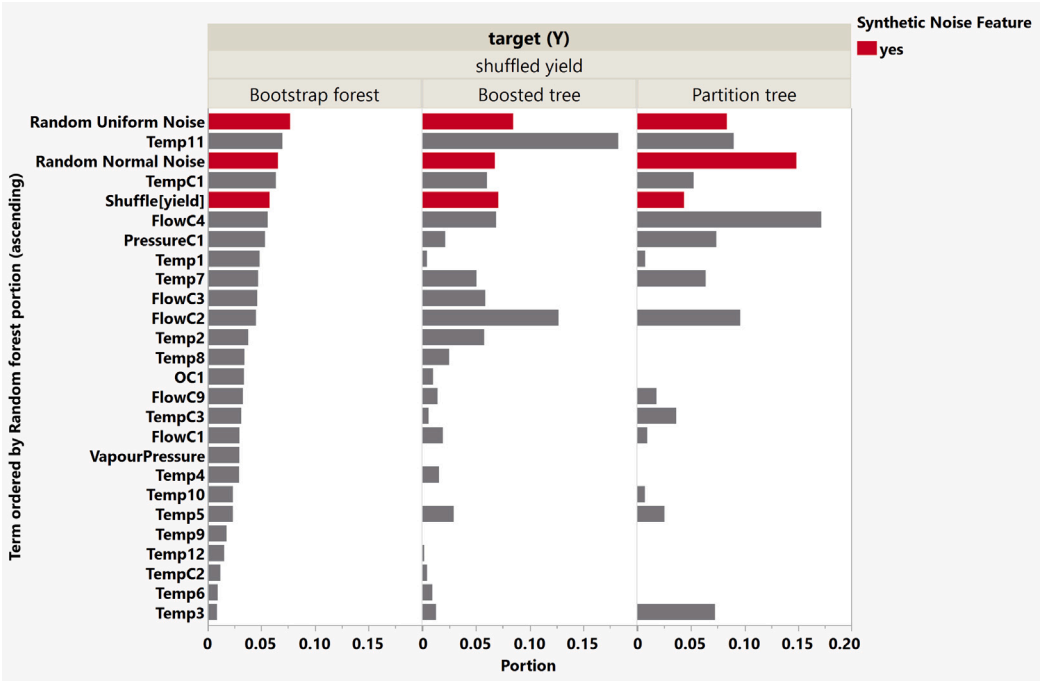

Fig. 3. When product-yield values are randomly shuffled, the synthetic-noise features leap to the top of the importance chart, immediately signaling that no real process variable carries predictive power. Features are ranked by their relative contribution to the total fitted variation explained by the model. Feature-importance bar plot for the distillation-column dataset with shuffled yield; SNF bars (red) dominate every model, confirming the absence of explanatory signal in the original features.

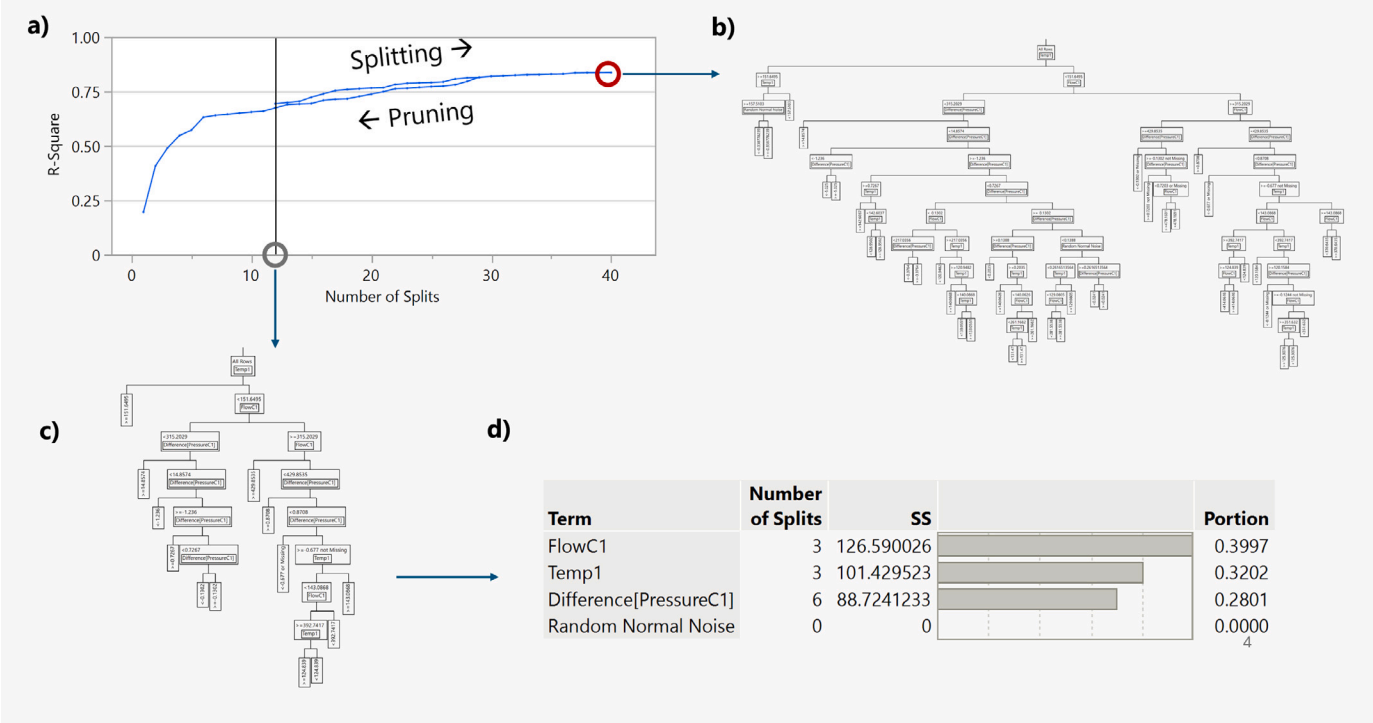

Fig. 4. The first split that selects the synthetic-noise feature (SNF) pinpoints the exact depth at which the regression tree begins to overfit, providing an objective pruning rule. Stopping a regression tree with an SNF: **(a)** Training R^2 vs. number of splits. The first SNF split (gray circle at 12 splits) defines the *noise threshold*, whereas further splits (blue arrow toward the red circle at ≈40 splits) raise R^2 also by fitting noise. **(b)** Fully grown tree obtained by ignoring the threshold and splitting until R^2 is maximized. **(c)** Tree pruned back to the last *signal* split, i.e., the split just before the SNF would have been used. **(d)** Variable-importance summary for the pruned tree: the three process variables receive all split counts, while the SNF registers zero influence, confirming that the stopping rule prevents the model from learning noise.

non-informative feature, i.e., the SNF, to explain the target variable. Therefore, to avoid overfitting, all splits made by the algorithm after that point should be pruned, including the first SNF split. This idea leads to the development of self-regularizing ML methods that take this indication into account and automatically yield the most parsimonious model in the training step.

The same concept can be illustrated for an Artificial Neural Network (ANN) model, see Fig. 5. In this case, controlling model complexity is harder because ANNs are difficult to interpret. JMP (Statistical Discovery) model profiler or a partial dependence plot can be used. This interface shows the effect of varying only one of the model inputs on the target variable, see Fig. 5**(c)** and **(d)**. The best indication in this case is the shape of the response curve between the SNF feature and the target. If the curve deviates from a flat response, it indicates that the ANN has attributed a non-zero effect to the SNF feature. The model is then using noise to explain and predict the target variable. This resulting model is now overfitting, and therefore, the number of nodes in the ANN structure can be reduced. Repeat this pruning until the response curve between the SNF and the target is close to zero in all the desired feature spaces.

This noise-based pruning is closely related to *dropout* (Srivastava et al., 2014) (randomly deactivating units during training) and to weight *regularization* (penalizing large weights) (Lu et al., 2018; Shen et al., 2024); together these belong to the broader family of neural-network model reduction/compression methods used to curb overfitting and simplify deployment.

3. Unsupervised learning

In this section, SNFs are used in unsupervised machine learning problems. This class of problems is particularly useful in a process-industry setting, where monitoring hundreds or thousands of sensors or key performance indicators is required. This class of problem is the cornerstone of what in the process industry is commonly referred to as Predictive Maintenance and/or Anomaly Detection applications. For this purpose, we will first motivate the need for this methodology and introduce the industrial datasets used in this section.

3.1. Motivation

Unsupervised Learning is mainly used in industrial practice for anomaly detection purposes. For example, yield, throughput, and quality are constantly monitored using classical statistical techniques taught in Six Sigma courses, Montgomery and Woodall (2008), Palací-López et al. (2020). For instance, control charts can be viewed as a simple unsupervised method: they do not distinguish inputs/outputs but flag in-control vs out-of-control behavior (Mowbray et al., 2022).

In the case of having multiple metrics that move in the same direction, dimensionality reduction aims at reducing the feature space by aggregating different features in latent variables. This aggregation can be done both linearly, with Principal Component analysis (PCA), or non-linearly (KPCA, UMAP), depending on the selected algorithm, see Schölkopf et al. (1997) and Joswiak et al. (2019), respectively. In practice, this is implemented via PCA-based multivariate statistical process control (MSPC) — including its multiway/batch extensions — with numerous real industrial case studies; see Nomikos and MacGregor (1995), García-Muñoz et al. (2003), García-Muñoz (2004), Ferrer (2014), MacGregor et al. (2005).

Analogous to the previous section, we add synthetic noise features (SNFs) to the feature set and use them as an internal reference to determine the number of latent variables. The presented approach is related to, but distinct from, Horn's parallel analysis, which is estimated from separate random datasets (Horn, 1965) or via permutations (Buja and Eyuboglu, 1992; Vitale et al., 2017).

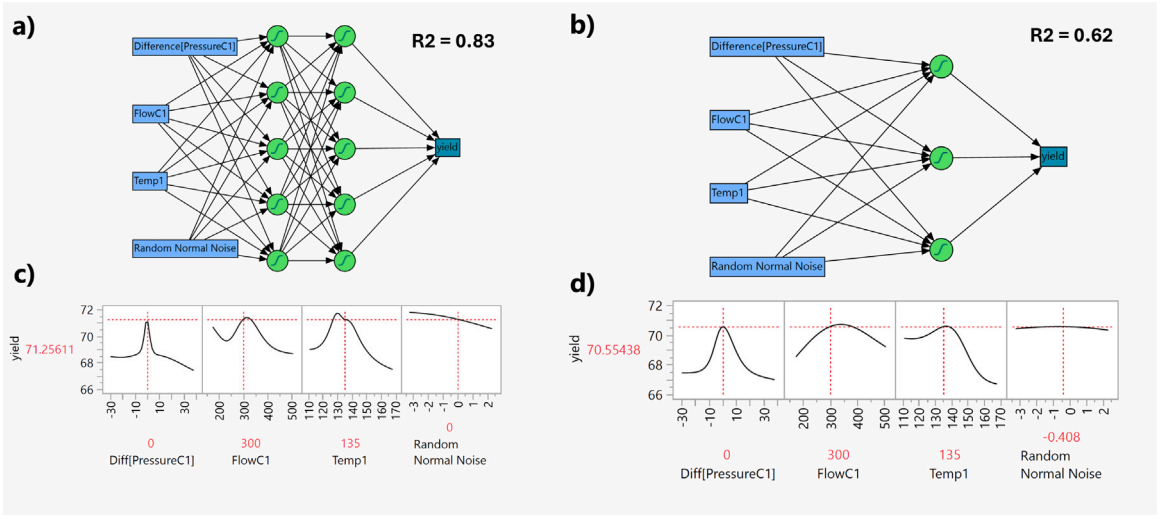

Fig. 5. A non-flat partial-dependence curve for the synthetic-noise input reveals that the original ANN is over-parameterized and guides its compression to a simpler, more robust model. Using an SNF to regularize an ANN predictor: **(a)** Over-sized network with two hidden layers ($R^2 = 0.83$); its excess capacity assigns non-zero weights to the SNF (input labeled *Random Normal Noise*). **(c)** Model profiler for (a) shows a clear response to the SNF, confirming that the network is fitting noise. **(b)** Parsimonious network obtained by reducing the architecture to one hidden layer of three neurons ($R^2 = 0.62$); the lower capacity prevents reliance on the SNF. **(d)** Profiler for (b) now displays an almost flat curve for the SNF, demonstrating that the streamlined model no longer uses noise to explain variation in product-yield.

3.2. Unsupervised learning datasets

Two datasets will be used in this section. The first is a minimal example with two variables (features): the oil fraction in an oil–water mixture and its density (Mowbray et al., 2022). It is clear how these two variables are correlated with each other; the more oil is present in the mixture, the lower its density will be. This example will be used to illustrate the proposed methodology. The second dataset is generated using the well-known Tennessee Eastman Plant (TEP) model, Downs and Vogel (1993). This dataset includes 22 variables, and it will be used to illustrate the application of the proposed methodology to an industrially relevant anomaly detection problem.

3.3. Dimensionality reduction

One of the main decisions that a user needs to take when applying methods for dimensionality reduction and/or multivariate correlation analysis is the identification of how many latent variables should be retained. This is true for all methods, linear and non-linear, and it is a trade-off between more variance explanation (i.e., more latent variables) and more robustness (i.e., fewer latent variables). In this paper, we propose applying the SNFs methodology to make an informed decision on how many latent variables to retain and guard against spurious latent dimensions. In particular, the proposed methodology discounts all Principal Components (PCs) after and including the first one, where an SNF is the main variable used to explain the variance in that component.

3.4. Intuitive example

The methodology will be shown with an illustrative dataset introduced above by applying PCA to perform a dimensionality reduction. Fig. 6 shows the data included in the illustrative example dataset with three control charts, see Fig. 6. Panels **(a)** and **(b)** show the data of the two real variables while **(c)** shows the added SNF. Fig. 6**(d)** shows a scatterplot of the oil fraction vs. the density and the first and second PCs obtained from applying PCA on this dataset. It is clear from the scatter plot that oil fraction and density are highly correlated, as explained in 3.2. Fig. 7 shows the absolute loadings for each variable in each PCs obtained for the illustrative example. PC1 shows high loadings for the two actual variables and low ones for the added SNF, with a 53.9% of explained variance. PC2, on the other hand, shows high loadings for the added SNF, low loadings for the two actual variables, and a 34.3% of explained variance. Finally, PC3 shows lower loadings and an 11.8% of explained variance. Following the proposed methodology, one should only keep PC1 and discard PC2 and PC3. The same information is visualized in Fig. 8. In this case, the PCs are color-coded, and the original variables plus the SNF appear on the y-axis. As can be seen, the same conclusion can be reached from this plot. It is also clear how the SNF is mainly used in PC2, and that PC3 only carries a very marginal explanatory variance power.

3.5. Industrial example: Tennessee Eastman plant

The proposed methodology can be applied to more complex and industrially relevant datasets. This is showcased on the TEP use case and dataset presented in sub- Section 3.2. Fig. 9**(a)**, which was adapted from Fortela and Mikolajczyk (2023), represents a P&ID of the TEP plant, while 9**(b)** shows the trend of all variables present in the dataset. As can be seen, the dataset shows a dynamic response of the TEP plant to a load change in the component A Feed, Stream 1 (S1), to the Reactor. This change affects most of the variables in the plant, and eventually, due to the PIDs present in the simulation, a new steady state operating point is reached.

Also, in this case, we demonstrate the methodology applied to a PCA analysis. However, it should be noted that it can also be applied to UMAP, McInnes et al. (2018), Joswiak et al. (2019), or any other dimensionality reduction method. Fig. 10 **(a)** shows the cumulative% of explained variance and the individual% for each PC. PC5 and after are highlighted in red since the proposed methodology has identified them to be non-relevant. PC5 is the first PC component where the introduced SNF is the most relevant variable. This is seen in Fig. 10**(b)**. This plot shows a heatmap graph between the features used for the analysis on the y-axis and the principal components obtained from the analysis on the x-axis. The heatmap is colored based on the absolute loadings of each feature in each component. As can be seen, the absolute loading of the SNF feature first becomes relevant for PC5, where it is the feature with the highest absolute loading, while for PC1 to PC4, this is near 0. Therefore, according to the proposed methodology, all PCs after and including PC5 are deemed to be non-relevant and should not be

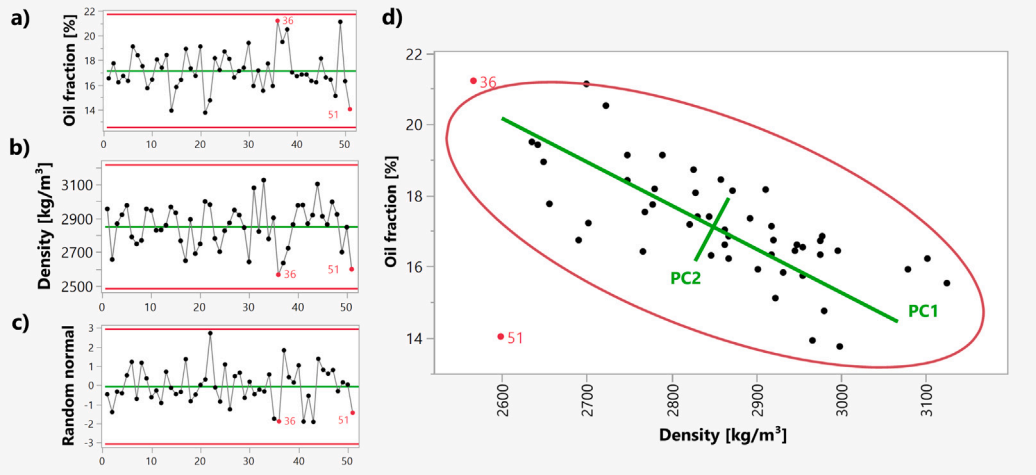

Fig. 6. Control charts show the noise trace as unstructured white noise, whereas oil fraction and density move coherently. Control charts for **(a)** oil fraction, **(b)** Density and **(c)** Synthetic Noise Feature (SNF). **(d)** shows the scatter plot of oil fraction vs. density and highlights the Principal Component (PC) 1 and PC2.

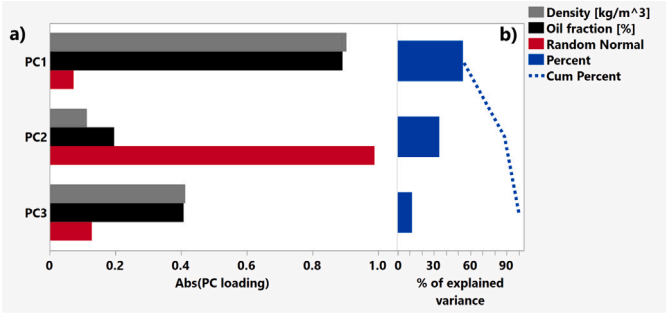

Fig. 7. Absolute loadings show that the synthetic-noise feature dominates PC 2, signaling that only the noise-free PC 1 should be retained for further analysis. Demonstration of SNF-guided dimensionality reduction on an illustrative dataset: **(a)** PCA absolute loadings for the three variables across the first three principal components; **(b)** percentage of variance explained by each component.

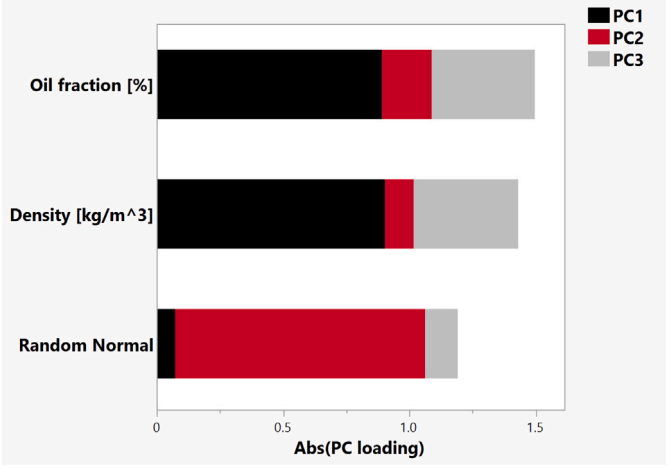

Fig. 8. Stacked cumulative loadings show that PCs 2 and 3 are driven almost entirely by the synthetic-noise feature, confirming that only PC 1 carries a real signal under the SNF cutoff. Cumulative sum of PCA absolute loadings for the three variables across the first three principal components in the illustrative dataset, with bar segments color-coded by contributing PC.

retained for further analysis. In the same way, it can also be observed that a subset of features becomes relevant only in PC5 to PC8, but it is not in PC1 to PC4. Analogously, it can also be concluded that these features carry low information, at least lower or of the same order of magnitude carried by the added SNF. Therefore, this methodology also allows not only for PC selection but also for feature selection in unsupervised dimensionality reduction applications. Fig. 11 illustrates the same concept more concisely. The plot reports the cumulative sum of the absolute loading per variable. Different colors indicate which portion belongs to which PCs. The SNF is the dominant variable in PC5, together with the stripping level, the recycle flow (S8), and the Reactor Feed. The variables are visualized top to bottom according to the cumulative sum of the absolute loadings of PC1 to PC5 included. As can be seen, all variables below the SNF feature show a lower cumulative sum of absolute loadings with respect to the rest. This leads to the same conclusion that we have reached by looking at plot 10. These variables can therefore be discarded since they are not relevant in this analysis. Finally, only PC1 to PC4 should be retained for further analysis and use.

3.6. Anomaly detection

Anomaly detection is a core unsupervised-learning task in the process industries. As discussed in 3.1, the prevailing workflow builds a multivariate *normal-operation* reference from historical data and raises an alert when new observations deviate from that baseline; this baseline–deviation logic underpins many condition monitoring software and algorithms (Mowbray et al., 2022; Bouman et al., 2024). The same problem can also be cast as supervised: assign labels to time segments (e.g., months or batches) and train a discriminative model to separate them. This could be done either in a change-point framework that optimizes a classifier-based objective, Londschien et al. (2023), or as a classifier two-sample test, Hediger et al. (2022). In both cases, standard variable-importance measures reveal which sensors drive the observed change.

The TEP example (see Fig. 9) is also used to showcase an extension of this approach. The main idea is to use an independent monotone increasing variable, such as time or the row index in the dataset table, as the target variable. This implies that the supervised learning method will try to find variables that can explain the variation of the target variable. Fig. 12**(a)** shows this concept applied to the TEP dataset, and in this case, the row index was selected as the independent target variable. This shift allows using the same SNF approach described in Section 2.3. As can be seen from Fig. 12**(a)** and **(b)**, where the added SNF acts as a threshold for determining which of the selected

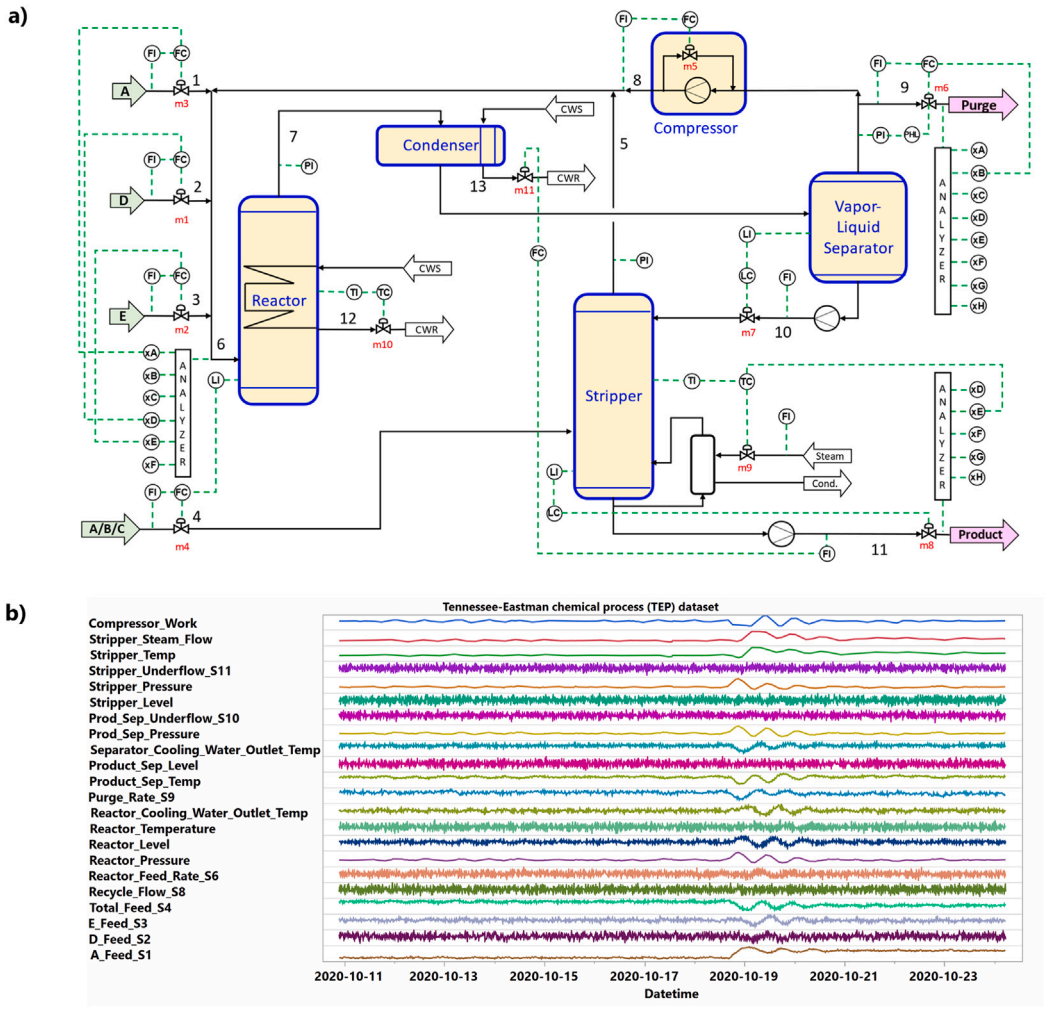

Fig. 9. Visualization of the TEP use case. **(a)** P&ID of the TEP plant (Fortela and Mikolajczyk, 2023) and **(b)** trends of some of the variables included in the dataset. The full dataset is available at Vallerio and Navarro-Brull (2025).

features is the most anomalous in the considered period. The variables are correctly ranked, based on the resulting feature importance, from the most anomalous (i.e., highest variation) to the least anomalous (i.e., lowest variation). The SNF acts as a visual cut-off in figure **(a)** and a quantitative one in **(b)**.

4. Active learning

The previous sections have shown how adding Synthetic Noise Features (SNFs) provide an intuitive approach for supervised and unsupervised analytics. We now turn to a question that matters even in the case of pilot plants or recipe optimization for new products:

> "When experimentation is expensive, can we learn and optimize a process in real time without wasting runs exploring irrelevant factors?"

The traditional playbook in chemical-process development is a **two-stage Design of Experiments (DoE) sequence**:

1. **Screening stage.** A fractional factorial or, more recently, a Definitive Screening Design (DSD) is run to separate the *few that matter* from the *many that do not*, Montgomery (2008). Small-effect and inert variables are discarded, simplifying both the model and the process recipe.
2. **Response-surface stage.** With the factor list pruned, a curvature-friendly layout — central composite, Box–Behnken, or an I-optimal custom design — maps interactions and quadratic terms so that an optimum can be located and verified, Jones et al. (1998).

While this workflow offers statistical guarantees with minimum experimentation, it remains a *batch, one-shot strategy*: every run in each stage must be scheduled before any data is seen, leaving no room for mid-course corrections once surprises emerge on the plant floor. Moreover, the primary objective of classical DoE is to *explore and map* the system response over the factor space, rather than to directly locate an optimum.

Active learning (AL)[1] removes these constraints by adaptively selecting new experiments based on an updated model that assimilates

[1] Of which Bayesian optimization is a subfield, Shahriari et al. (2016), Frazier (2018).

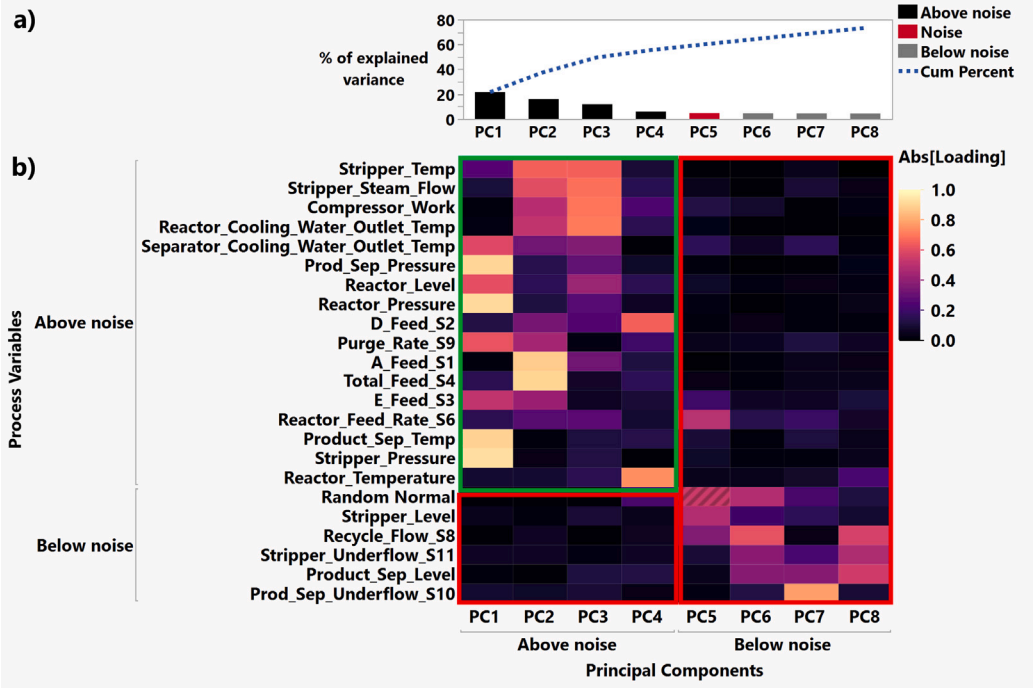

Fig. 10. PC 5 emerges as the first principal component dominated by the synthetic-noise feature, establishing a clear cutoff for dimensionality reduction. **(a)** Cumulative and individual percentages of explained variance for each principal component in the Tennessee Eastman dataset. **(b)** Heat-map of absolute loadings (features on the *y*-axis, PCs on the x-axis) showing the synthetic-noise feature overtaking the real variables from PC 5 onward.

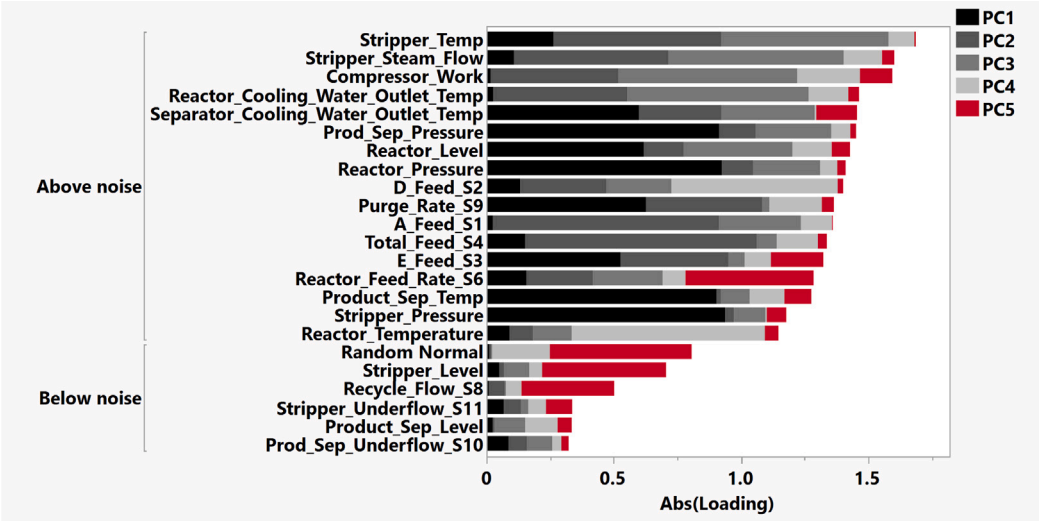

Fig. 11. Ranking variables by cumulative PCA loading relegates the synthetic-noise tag — and other low-impact sensors — below the cutoff while spotlighting the key TEP drivers. Shown is the cumulative sum of PCA absolute loadings for all variables across the first five PCs in the TEP dataset, with bar segments color-coded by the fractional contribution of each PC.

the latest data, thereby transforming a batch workflow into a closed-loop cycle whose objective is to efficiently *identify optimal operating conditions*.

The next subsections explain the differences between DoE and AL workflows, highlighting the role of SNFs used as an *in-situ* variable-selection signal. DoE is a well-established field; readers seeking an introductory overview are referred to Leardi (2009), while contemporary discussions on its relationship with machine learning can be found in Arboretti et al. (2022b,a).

4.1. Motivation and dataset

Chemical plants rarely allow hundreds of *try-and-see* trials; a single pilot run has a high cost in solvents, utilities, and downtime. To illustrate how DoE compares to AL, we adopt a three-factor variant of the Rosenbrock valley. This benchmark mimics the multifactor, highly curved response surfaces typical of separation and polymerization units.

The raw Rosenbrock response varies three orders of magnitude, substantially more than typical process data, so we first take a log

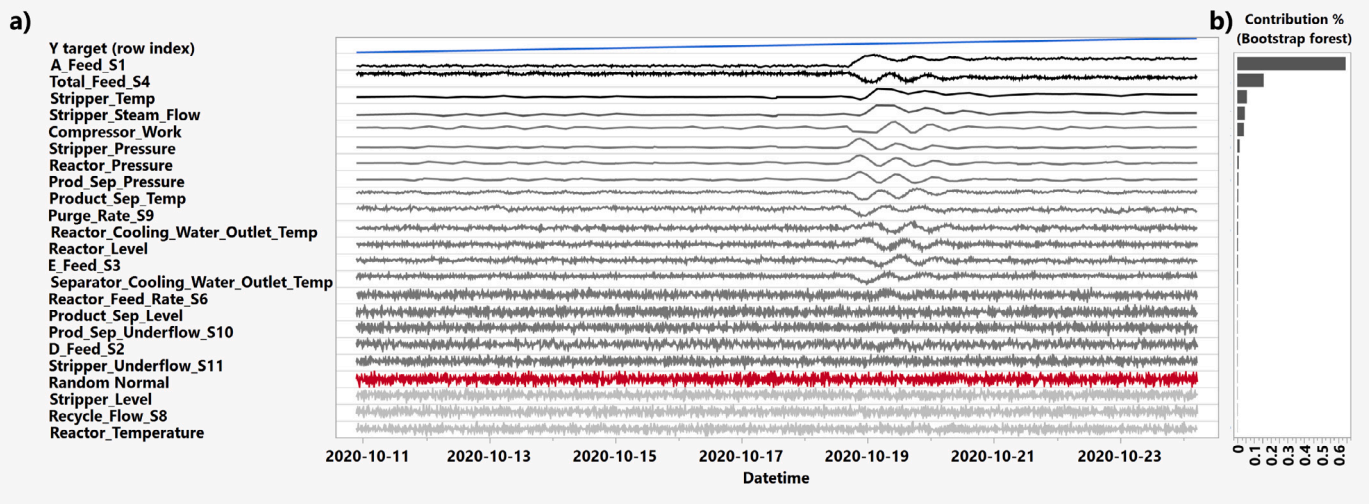

Fig. 12. The synthetic-noise feature sets a clear visual and quantitative cutoff, ranking truly abnormal variables above the noise floor in supervised anomaly detection. Demonstration on the Tennessee Eastman dataset: **(a)** row index used as an independent target alongside the full set of process variables; **(b)** feature-importance bar chart from a bootstrap-forest model, with the SNF (red) marking the threshold between informative and non-informative sensors.

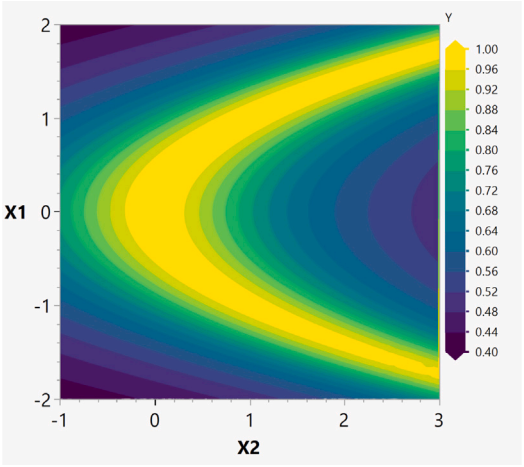

Fig. 13. Normalized Rosenbrock response surface used as a benchmark for comparing classical DoE and active learning. Shown is the two-dimensional slice in (X_1, X_2) highlighting the characteristic curved valley of optimal responses (brighter colors indicate higher objective values). A third factor, X_3, is included in the study but is intentionally non-informative and therefore not visualized here.

to compress the scale. We then normalize it using one of two simple options: (i) a min–max rescale to 0–1 across the relevant dataset (used in the DoE examples), or (ii) a reciprocal mapping $1/(\log \text{term} + 1)$ (used in active learning), which also keeps values in (0,1] and further damps outliers. In both cases we keep the *higher is better* direction (see Fig. 13). The global optimum remains located at $(0, 0)$, although these transformations flatten the response surface near the optimum and therefore make convergence more challenging.

$$\underbrace{f(X_1, X_2)}_{\text{Rosenbrock}} = (1 - X_1)^2 + 100\,(X_2 - X_1^2)^2, \qquad Y = \log_{10}\big(f + 1\big). \tag{1}$$

A third factor X_3 is inserted on purpose as a *non-informative* variable (Linkletter et al., 2006), playing a similar role that SNFs played in Sections 2–3, in this case using Active Learning and monitoring variable importance (Savitsky et al., 2011).

For clarity, the results reported here omit additive and multiplicative noise on the response. The repository, Vallerio and Navarro-Brull (2025), includes notebooks that enable both noise modes and replicate the study on additional function benchmarks (Booth, anisotropic sphere, Hosaki, Himmelblau's, and the Three-Hump Camel), along with all code and generated datasets.

For clarity, the results reported here omit additive and multiplicative measurement noise on the response. The repository (Vallerio and Navarro-Brull, 2025) includes notebooks that enable both noise modes and replicate the analysis on additional benchmark functions (Booth, anisotropic sphere, Hosaki, Himmelblau's, and the Three-Hump Camel), together with all code and generated datasets.

4.2. Design of Experiments – DoE

Classical factorial and composite designs place most of their points on the *corners* and *edges* of the experimental cube (or simplex). Modern implementations of this approach are, for example, Definitive Screening Designs (DSD), Jones and Nachtsheim (2011).

That strategy is deliberate: (i) to sample the extremes of the factor space in order to maximize information content with the fewest runs, which is critical when each experiment consumes hours of reactor time or raw material; and (ii) to allow a response-surface model (RSM) to capture curvature, under the assumption that the true surface remains close to quadratic. In this context, interaction effects refer to situations in which the influence of one factor on the response depends on the level of another, and are commonly represented by multiplicative terms between variables.

Two caveats matter in everyday plant work:

1. **Strong non-linearity.** If the real response folds or twists beyond a quadratic shape, curvature estimates become hypersensitive to which factors define the design's edges. A few unlucky corner points can mislead the entire model.
2. **Mixture or compositional systems.** In formulation problems (paints, solvents, polymer blends), the *corners* represent pure components that will never be sold as products. Those extreme blends add cost but little insight. Modern practice therefore favors *space-filling* layouts inside the feasible region: Latin hypercubes (Viana, 2016), maximin designs, or optimal *inside-simplex* algorithms, Cornell (2011).

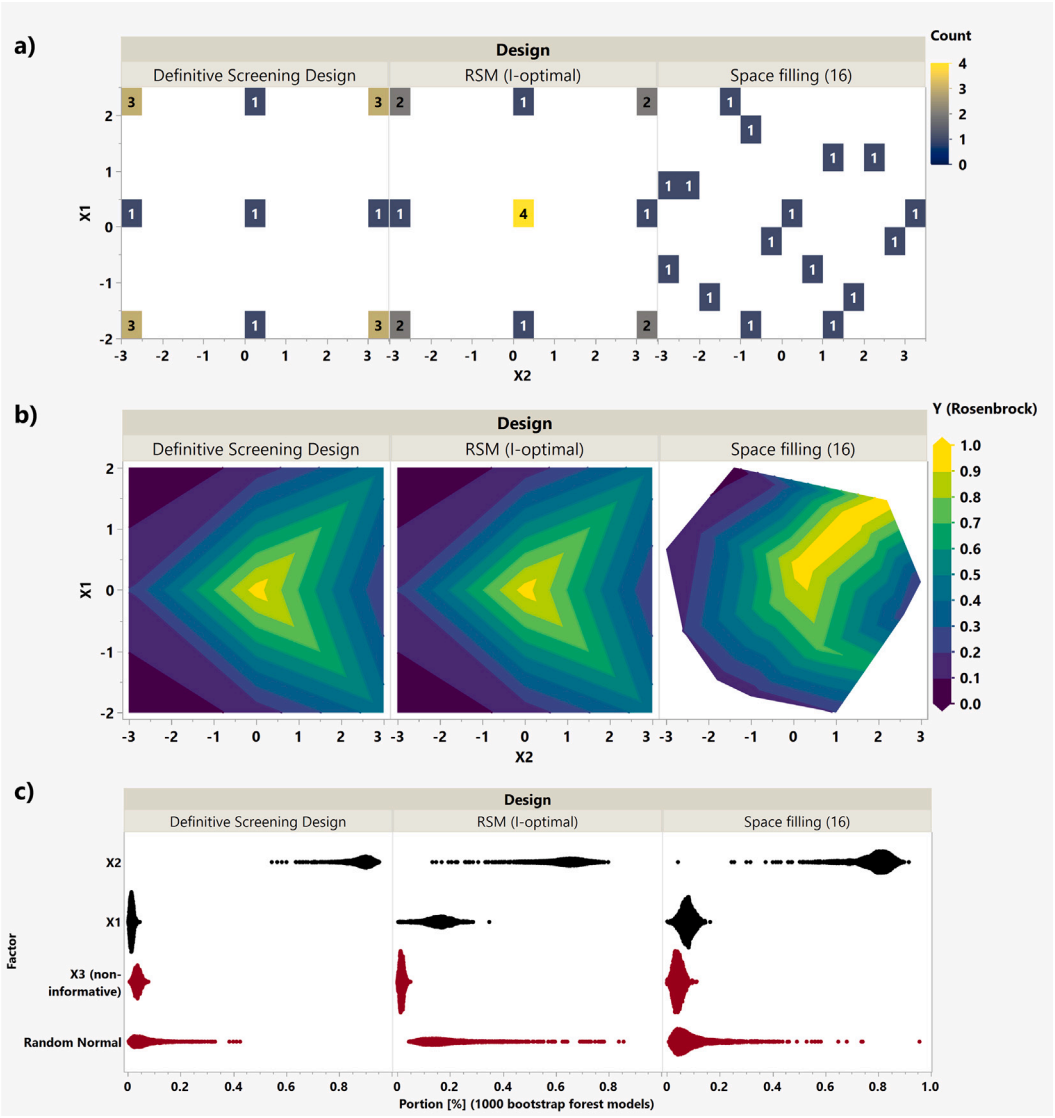

Fig. 14. This benchmark illustrates how space-filling designs can reveal global structure and inert factors that remain poorly resolved by classical DoE at the same run budget. **(a)** Point layouts for a Definitive Screening Design (DSD), an I-optimal response-surface design (RSM), and a space-filling Latin hypercube (LH), projected onto the X_1–X_2 plane (colors indicate replicate counts). **(b)** Contours of the scaled Rosenbrock response; only the space-filling design resolves the curved banana-shaped optimum with fidelity. **(c)** Factor-importance distributions from 1 000 bootstrap random-forest models with a regenerated Random Normal SNF; the space-filling design most clearly demotes the non-informative factor X_3 to the noise floor.

Capabilities and limitations of classical doe. Fig. 14 summarizes the strengths and limitations of classical DoE. With the same 16 to 20 run budget, a DSD and an I-optimal RSM both cluster around the center and axial points—ideal for estimating curvature, yet poor for mapping the global shape (see Fig. 14a). The number of runs in each design is the minimum required by the selected design to estimate higher-order effects. The space-filling Latin hypercube scatter, by contrast, *does* expose the banana-shaped valley (Fig. 14b) and crucially pushes the bootstrap importance of the non-informative X_3 below that of the SNF (Fig. 14c) The message echoes Section 2.3:

- A single space-filling or I-optimal design, with well-chosen corner points, is usually sufficient to (i) detect non-linearities, (ii) flag inert factors, and (iii) obtain an initial surrogate model.
- Once the optimality valley is hinted at, adding another evenly spread batch yields little new information; a targeted, sequential strategy is far more cost-effective.
- Because a dataset this small is statistically fragile, we refit the random-forest screener **1 000** times, each time regenerating the SNF. The bootstrap cloud in Fig. 14**(c)** reveals how often a purely random feature can outrank a real factor, the same safeguard used in the supervised-learning study (Section 2.3).

4.3. Active learning: Exploration versus exploitation

Design of Experiments treats the test matrix as a fixed, upfront plan: all corner and center experiments are defined before any data are collected, under the assumption that a quadratic approximation will adequately describe the response surface.

Active learning removes this constraint. Rather than committing to the full experimental plan at the outset, an initial DoE seed is executed, a surrogate model is fitted, and subsequent experiments are selected sequentially based on the updated model. Each new experiment can be chosen to prioritize expected improvement (*exploitation*), information gain (*exploration*), or a balance of both.

After each experiment, the model is updated, uncertainty is re-evaluated, and the cycle repeats until the optimization objective is achieved or the experimental budget is exhausted.

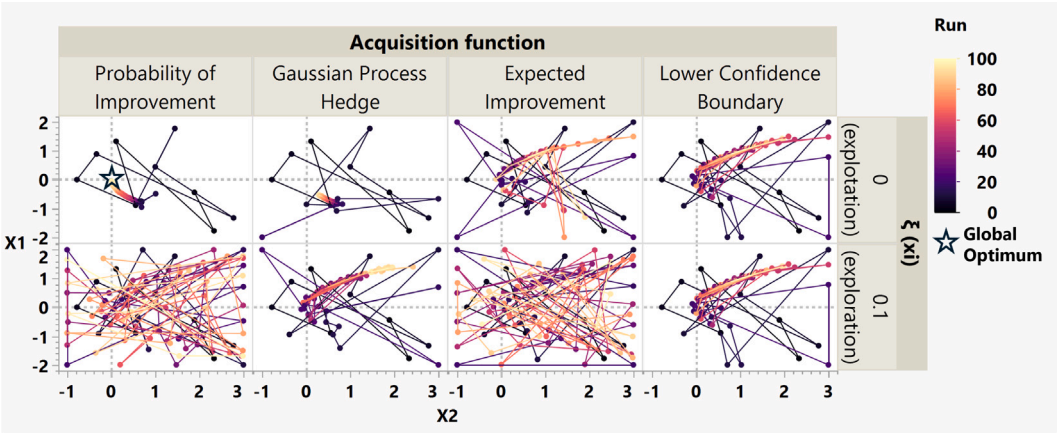

Fig. 15. Active-learning trajectories are highly sensitive to the choice of acquisition function and exploration bias. Search paths are shown for four common acquisition functions—Probability of Improvement (PI), Expected Improvement (EI), GP-Hedge, and Lower Confidence Bound (LCB)—under two exploration settings: $\xi = 0$ (top, pure exploitation) and $\xi = 0.1$ (bottom, mild exploration).

For chemical engineers, the appeal is two-fold:

- **Fewer wasted runs.** Each experiment is conducted with the benefit of all data collected so far, so effort naturally migrates toward profitable or poorly understood regions instead of filling out a geometric pattern.
- **Live variable selection.** Data-driven models equipped with variable importance monitoring acting like a real-time SNF test: if a factor proves inert, we can freeze or drop that variable.

4.4. Practical limitations of active learning and the role of synthetic noise awareness

Bayesian optimization and active-learning frameworks are now established tools in both industrial and open-source ecosystems, including Merck's BayBe (Fitzner et al., 2025), Novo Nordisk's Process Optimizer (Bertelsen et al., 2025), and BASF's bofire (Dürholt et al., 2024). These platforms implement *sequential* experiment selection, in which each new experiment is proposed using all information gathered so far, rather than committing to a fixed experimental matrix in advance. Any surrogate model capable of providing predictive uncertainty — such as Gaussian processes, random-forest ensembles, or Bayesian neural networks — can be used within this framework (Rasmussen and Williams, 2005; Frazier, 2018).

Despite their conceptual appeal, two practical issues limit the robustness of turnkey active-learning deployments in industrial settings:

1. **Sensitivity to acquisition-function choice.** Acquisition functions mediate the trade-off between exploration and exploitation. In practice, different acquisition functions—and even small changes in their exploration bias—can steer an experimental campaign along markedly different trajectories.
2. **Persistence of non-informative factors.** Active-learning algorithms may continue to perturb variables (input factors) that contribute little or no information. This behavior increases process variability and complicates operator interpretation.

To illustrate these effects, multiple scenarios with different acquisition functions and exploration parameters were simulated. Each scenario aggregates 50 independent simulations, all initialized from the same 9-point Latin hypercube design and run for 91 active-learning iterations (100 total experiments).

Fig. 15 shows that all acquisition strategies eventually locate the curved optimum valley, yet the path taken — and the associated process variability — depends strongly on both the acquisition function and the exploration bias.

A complementary perspective is provided by examining the evolution of Gaussian-process length scales. Fig. 16 shows that, after an initial exploration phase, length-scale dynamics provide an early indication that certain input dimensions are non-informative.

An interactive noise filter. Even under idealized conditions with perfectly observed responses, non-informative variables continue to be explored for all acquisition functions and despite being under pure exploitation. This is illustrated in Fig. 17 where the experimentation was continued.

To address this issue, a supervised-learning–based workflow can be adopted, analogous to the approach described in Section 2.3:

1. **Initial screening.** Perform a minimal Definitive Screening Design (DSD) or Latin hypercube (LH) design augmented with a synthetic noise feature (SNF), and discard any factor whose importance overlaps with that of the SNF.
2. **Monitoring of non-informative dimensions.** During active learning, retain a known non-informative reference input and monitor Gaussian-process length scales. Any variable whose length scale grows to the same order of magnitude as that reference can be frozen, as it no longer contributes meaningful information.

In summary, synthetic noise features provide not only interpretability but also a practical control mechanism for active learning. By explicitly identifying and managing non-informative factors, they allow engineers to exploit the efficiency of sequential experimentation without unnecessary exploration of irrelevant dimensions.

5. Conclusions

Chemical engineers face a persistent challenge: modern plants record thousands of tags, yet the time and budget available for modeling and experimentation are limited.

This paper demonstrates that *injecting* one or more Synthetic Noise Features (SNFs) into the dataset converts that paradox into an advantage.

- In **supervised learning**, an SNF establishes a visual and quantitative threshold for variable importance. Any predictor less informative than pure noise is pruned automatically, and the

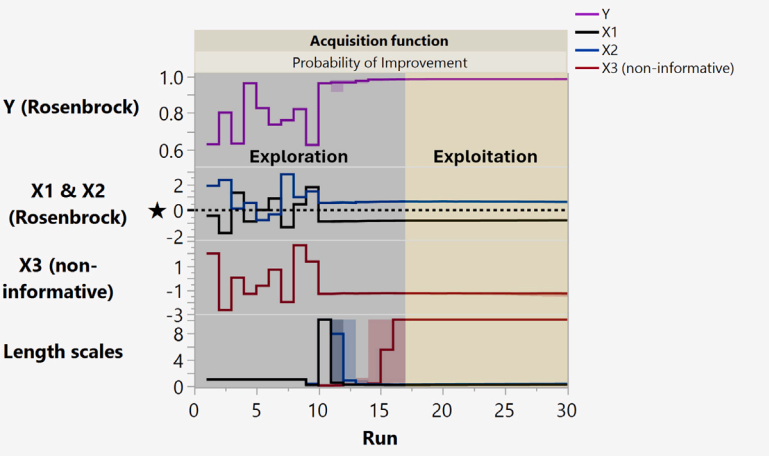

Fig. 16. Gaussian-process length scales provide an early diagnostic of non-informative inputs during active learning. Following the initial exploration phase, the length scale associated with the dummy variable X_3 rapidly saturates at the upper prior bound, indicating practical irrelevance, while the informative variables X_1 and X_2 remain well constrained. Solid lines show the median trajectory and shaded bands the interquartile range, aggregated over 50 independent simulations.

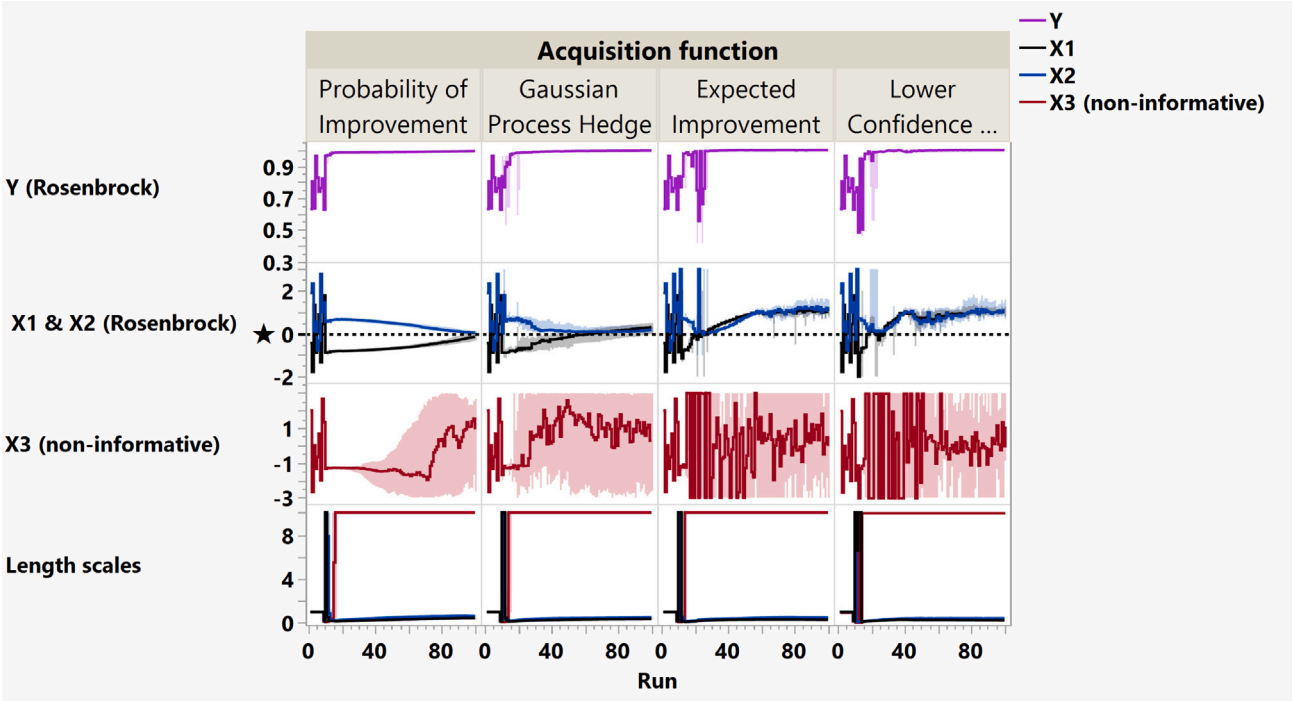

Fig. 17. If length-scale diagnostics are ignored, active learning continues to explore non-informative variables despite no additional information gain. As shown in the preceding figure, the Gaussian-process length scale associated with the dummy factor X_3 rapidly saturates at the upper prior bound after ~20 runs, indicating practical irrelevance. Nevertheless, most acquisition functions continue to perturb this dimension after 30 runs, even under pure exploitation and in the absence of measurement noise. Solid lines show the median trajectory across repeated simulations; shaded bands indicate variability (interquartile range).

same signal flags overgrown trees or overparameterized neural networks for pruning and regularization.

- In **unsupervised learning**, the SNF marks the first principal component that should be rejected. Latent variables dominated by noise are discarded, yielding more robust PCA models and cleaner anomaly detection dashboards.
- In **active learning**, the SNF, implemented as a non-informative factor, works in tandem with GP length scales to halt wasted exploration. Screening once, then freezing factors whose length scale explodes, may cut the experimental budget and reduce unnecessary excitation on the studied process.

Because SNFs require no changes to existing algorithms, the method can be applied directly to any already available ML method, independently of the class of problem. The approach is therefore immediately actionable for recipe optimization, predictive maintenance,

and process-monitoring projects where model transparency and rapid iteration are paramount.

Decision support and responsibility. SNFs are a *decision-support* technique, not a substitute for engineering judgment. Data-driven analyses can mislead when applied outside validated operating windows (extrapolation risk) or in the presence of confounding; they can also fail for data-quality reasons—e.g., missing or sparse data, corrupted or mislabeled tags, malfunctioning or out-of-calibration instruments, inadequate sampling frequency, or poor LIMS–historian integration (timestamp/sample-ID misalignment, unmerged data sets). SNFs reduce these risks by providing a transparent threshold and by flagging uninformative signals, thus avoiding premature preselection of a narrow set of suspected causes. However, they do not replace hazard reviews or process expertise. Sound decisions still require human judgment in the *chemistry*, the *process/unit operations*, and the *control strategy* (loops, interlocks, APC/MPC) to interpret model outputs and arrive at reasonable, safe actions.

Outlook. Future work will extend this approach to dynamic (time-series) models and integrate SNF-based stopping rules directly into active learning algorithms.

CRediT authorship contribution statement

Mattia Vallerio: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Antonio del Rio Chanona:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Francisco J. Navarro-Brull:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Responsibility and safe use

The methods and examples presented in this manuscript are intended for decision support and research purposes only. The safe operation of chemical processes, including compliance with applicable regulations, safety standards, and operational constraints, remains the responsibility of the user. The authors do not assume responsibility for decisions or actions taken based on the application of the methods described herein.

Use of generative AI tools

The authors used generative artificial intelligence tools (ChatGPT, OpenAI) to assist with language editing, stylistic refinement, and code drafting during the preparation of this manuscript. All scientific ideas, methodological choices, analyses, and conclusions are original and were developed by the authors.

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.

References

Altmann, A., Toloşi, L., Sander, O., Lengauer, T., 2010. Permutation importance: A corrected feature importance measure. Bioinformatics 26 (10), 1340–1347. http://dx.doi.org/10.1093/bioinformatics/btq134, URL: https://academic.oup.com/bioinformatics/article/26/10/1340/193348.

Arboretti, R., Ceccato, R., Pegoraro, L., Salmaso, L., 2022a. Design choice and machine learning model performances. Qual. Reliab. Eng. Int. 38 (7), 3357–3378. http://dx.doi.org/10.1002/qre.3123, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.3123 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/qre.3123.

Arboretti, R., Ceccato, R., Pegoraro, L., Salmaso, L., 2022b. Design of experiments and machine learning for product innovation: A systematic literature review. Qual. Reliab. Eng. Int. 38 (2), 1131–1156. http://dx.doi.org/10.1002/qre.3025, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.3025. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/qre.3025.

Arzac, I.I., Vallerio, M., Perez-Galvan, C., Navarro-Brull, F.J., 2023. Industrial data science for batch reactor monitoring and fault detection. In: Machine Learning and Hybrid Modelling for Reaction Engineering: Theory and Applications. Royal Society of Chemistry, http://dx.doi.org/10.1039/BK9781837670178-00358.

Arzac-Garmendia, I., Vallerio, M., Perez-Galvan, C., Navarro-Brull, F.J., 2022. Industrial data science for batch manufacturing processes. URL: https://arxiv.org/abs/2209.09660 arXiv:2209.09660.

Barber, R.F., Candès, E.J., 2015. Controlling the false discovery rate via knockoffs. Ann. Statist. 43 (5), 2055–2085. http://dx.doi.org/10.1214/15-AOS1337.

Barber, R.F., Candès, E.J., 2019. A Knockoff filter for high-dimensional selective inference. Ann. Statist. 47 (5), 2504–2537, URL: https://www.jstor.org/stable/26784037.

Barber, R.F., Candès, E.J., Samworth, R.J., 2020. Robust inference with knockoffs. Ann. Statist. 48 (3), 1409–1431. http://dx.doi.org/10.1214/19-AOS1852.

Bellman, R., 1961. Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton, NJ.

Bertelsen, S., Carlsen, S., Furbo, S., Nielsen, M.B., Obdrup, A., Taaning, R., 2025. ProcessOptimizer, an Open-Source Python Package for Easy Optimization of Real-World Processes Using Bayesian Optimization: Showcase of Features and Example of Use. J. Chem. Inf. Model. 65 (4), 1702–1707. http://dx.doi.org/10.1021/acs.jcim.4c02240, Publisher: American Chemical Society.

Bi, J., Bennett, K., Embrechts, M., Breneman, C., Song, M., 2003. Dimensionality reduction via sparse support vector machines. J. Mach. Learn. Res. 3 (null), 1229–1243.

Blum, A.L., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artificial Intelligence 97 (1–2), 245–271. http://dx.doi.org/10.1016/S0004-3702(97)00063-5.

Bouman, R., Bukhsh, Z., Heskes, T., 2024. Unsupervised anomaly detection algorithms on real-world data: How many do we need? J. Mach. Learn. Res. 25 (105), 1–34, URL: http://jmlr.org/papers/v25/23-0570.html.

Breiman, L., 2001. Random Forests. Mach. Learn. 45 (1), 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 2017. Classification and Regression Trees. Chapman and Hall/CRC, http://dx.doi.org/10.1201/9781315139470, Reprint of the 1984 edition.

Buja, A., Eyuboglu, N., 1992. Remarks on parallel analysis. Multivar. Behav. Res. 27 (4), 509–540. http://dx.doi.org/10.1207/s15327906mbr2704_2, PMID: 26811132.

Candès, E., Fan, Y., Janson, L., Lv, J., 2018. Panning for gold: Model-x knockoffs for high-dimensional controlled variable selection. J. R. Stat. Soc. Ser. B Stat. Methodol. 80 (3), 551–577. http://dx.doi.org/10.1111/rssb.12265.

Chandrashekar, G., Sahin, F., 2014. A survey on feature selection methods. Comput. Electr. Eng. 40 (1), 16–28. http://dx.doi.org/10.1016/j.compeleceng.2013.11.024.

Chiang, L.H., Lu, B., Castillo, I., 2017. Big data analytics in chemical engineering. Annu. Rev. Chem. Biomol. Eng. 8, 63–85. http://dx.doi.org/10.1146/annurev-chembioeng-060816-101555.

Cornell, J.A., 2011. A Primer on Experiments with Mixtures. In: Wiley Series in Probability and Statistics Ser, John Wiley & Sons, Incorporated, URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470907443.

Downs, J.J., Vogel, E.F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17 (3), 245–255. http://dx.doi.org/10.1016/0098-1354(93)80018-I.

Duhn, K., 2025. Learning chemical engineering. URL: https://learnche.org. (Accessed 21 August 2025).

Dürholt, J.P., Asche, T.S., Kleinekorte, J., Mancino-Ball, G., Schiller, B., Sung, S., Keupp, J., Osburg, A., Boyne, T., Misener, R., Eldred, R., Costa, W.S., Kappatou, C., Lee, R.M., Linzner, D., Walz, D., Wulkow, N., Shafei, B., 2024. Bofire: Bayesian optimization framework intended for real experiments. URL: https://arxiv.org/abs/2408.05040 arXiv:2408.05040.

Ferrer, A., 2014. Latent structures-based multivariate statistical process control: A paradigm shift. Qual. Eng. 26 (1), 72–91. http://dx.doi.org/10.1080/08982112.2013.846093.

Fitzner, M., Š.ošić, A., Hopp, A.V., Müller, M., Rihana, R., Hrovatin, K., Liebig, F., Winkel, M., Halter, W., Brandenburg, J.G., 2025. BayBE: a Bayesian back end for experimental planning in the low-to-no-data regime. "Digital Discovery" 4, 1991–2000. http://dx.doi.org/10.1039/D5DD00050E.

Fortela, D.L.B., Mikolajczyk, A.P., 2023. Detecting plant-wide oscillation propagation effects of disturbances and faults in a chemical process plant using network topology of variance decompositions. Processes 11 (6), http://dx.doi.org/10.3390/pr11061747, URL: https://www.mdpi.com/2227-9717/11/6/1747.

Frazier, P.I., 2018. A tutorial on Bayesian optimization. URL: https://arxiv.org/abs/1807.02811 arXiv:1807.02811.

Friedman, J.H., 2000. Greedy function approximation: A gradient boosting machine. Ann. Stat. 29, 1189–1232.

García-Muñoz, S., 2004. Batch Process Improvement Using Latent Variable Methods (Ph.D. thesis). McMaster University, Hamilton, Ontario, Canada, Open Access thesis URL: http://hdl.handle.net/11375/6274.

García-Muñoz, S., Kourti, T., MacGregor, J.F., Mateos, A.G., Murphy, G., 2003. Troubleshooting of an industrial batch process using multivariate methods. Ind. Eng. Chem. Res. 42 (15), 3592–3601. http://dx.doi.org/10.1021/ie0300023.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.

Hediger, S., Michel, L., Näf, J., 2022. On the use of random forest for two-sample testing. Comput. Statist. Data Anal. 170, 107435. http://dx.doi.org/10.1016/j.csda.2022.107435, URL: https://www.sciencedirect.com/science/article/pii/S0167947322000159.

Horn, J.L., 1965. A rationale and test for the number of factors in factor analysis. Psychometrika 30, 179–185. http://dx.doi.org/10.1007/BF02289447.

Jiang, T., Li, Y., Motsinger-Reif, A.A., 2020. Knockoff boosted tree for model-free variable selection. Bioinformatics 37 (7), 976–983. http://dx.doi.org/10.1093/bioinformatics/btaa770, arXiv:https://academic.oup.com/bioinformatics/article-pdf/37/7/976/50341175/btaa770.pdf.

Jones, B., Nachtsheim, C.J., 2011. A class of three-level designs for Definitive Screening in the presence of second-order effects. J. Qual. Technol. 43 (1), 1–15. http://dx.doi.org/10.1080/00224065.2011.11917841.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. J. Global Optim. 13, 455–492. http://dx.doi.org/10.1023/A:1008306431147.

Joswiak, M., Peng, Y., Castillo, I., Chiang, L.H., 2019. Dimensionality reduction for visualizing industrial chemical process data. Control Eng. Pract. 93, 104189. http://dx.doi.org/10.1016/j.conengprac.2019.104189.

Kursa, M.B., 2014. Robustness of random forest-based gene selection methods. BMC Bioinformatics 15 (1), 8. http://dx.doi.org/10.1186/1471-2105-15-8.

Kursa, M.B., Rudnicki, W.R., 2010. Feature selection with the boruta package. J. Stat. Softw. 36 (11), 1–13. http://dx.doi.org/10.18637/jss.v036.i11.

Leardi, R., 2009. Experimental design in chemistry: A tutorial. Anal. Chim. Acta 652 (1), 161–172. http://dx.doi.org/10.1016/j.aca.2009.06.015, Fundamental and Applied Analytical Science. A Special Issue In Honour of Alan Townshend. URL: https://www.sciencedirect.com/science/article/pii/S0003267009008058.

Li, J., Cheng, K., Wang, S., Morstatter, R.P., Trevino, R., Tang, J., Liu, H., 2017. Feature selection: A data perspective. ACM Comput. Surv. 50 (6), 94:1–94:45. http://dx.doi.org/10.1145/3136625.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., Ye, K.Q., 2006. Variable selection for Gaussian process models in computer experiments. Technometrics 48 (4), 478–490. http://dx.doi.org/10.1198/004017006000000228.

Londschien, M., Bühlmann, P., Kovács, S., 2023. Random forests for change point detection. J. Mach. Learn. Res. 24 (1).

Lou, H.H., Mukherjee, R., Wang, Z., Olsen, T., Diwekar, U., Lin, S., 2022. A new area of utilizing industrial internet of things in environmental monitoring. Front. Chem. Eng. 4, 842514. http://dx.doi.org/10.3389/fceng.2022.842514.

Lu, Y.Y., Fan, Y., Lv, J., Noble, W.S., 2018. Deeppink: reproducible feature selection in deep neural networks. URL: https://arxiv.org/abs/1809.01185 arXiv:1809.01185.

MacGregor, J.F., Yu, H., García Muñoz, S., Flores-Cerrillo, J., 2005. Data-based latent variable methods for process analysis, monitoring and control. Comput. Chem. Eng. 29 (6), 1217–1223. http://dx.doi.org/10.1016/j.compchemeng.2005.02.007, Selected Papers Presented at the 14th European Symposium on Computer Aided Process Engineering. URL: https://www.sciencedirect.com/science/article/pii/S0098135405000128.

McInnes, L., Healy, J., Melville, J., 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. https://arxiv.org/abs/1802.03426 arXiv preprint arXiv:1802.03426.

Montgomery, D., 2008. Design and Analysis of Experiments. John Wiley & Sons, URL: https://link.springer.com/book/10.1007/978-3-319-52250-0.

Montgomery, D.C., Woodall, W.H., 2008. An overview of six sigma. Int. Stat. Rev. 76 (3), 329–346. http://dx.doi.org/10.1111/j.1751-5823.2008.00061.x.

Mowbray, M., Vallerio, M., Perez-Galvan, C., Zhang, D., Del Rio Chanona, A., Navarro-Brull, F.J., 2022. Industrial data science – a review of machine learning applications for chemical and process industries. React. Chem. Eng. 7 (7), 1471–1509. http://dx.doi.org/10.1039/D1RE00541C.

Nomikos, P., MacGregor, J.F., 1995. Multivariate SPC charts for monitoring batch processes. Technometrics 37 (1), 41–59. http://dx.doi.org/10.1080/00401706.1995.10485888, URL: https://www.tandfonline.com/doi/abs/10.1080/00401706.1995.10485888 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/00401706.1995.10485888.

Ojala, M., Garriga, G.C., 2010. Permutation tests for studying classifier performance. J. Mach. Learn. Res. 11, 1833–1863, URL: https://www.jmlr.org/papers/volume11/ojala10a/ojala10a.pdf.

Palací-López, D., Borràs-Ferrís, J., da Silva de Oliveria, L.T., Ferrer, A., 2020. Multivariate six sigma: A case study in industry 4.0. Processes 8 (9), http://dx.doi.org/10.3390/pr8091119, URL: https://www.mdpi.com/2227-9717/8/9/1119.

Qin, S.J., 2014. Process data analytics in the era of big data. AIChE J. 60 (9), 3092–3100. http://dx.doi.org/10.1002/aic.14523.

Rasmussen, C.E., Williams, C.K.I., 2005. Gaussian Processes for Machine Learning. The MIT Press, http://dx.doi.org/10.7551/mitpress/3206.001.0001.

Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23 (19), 2507–2517. http://dx.doi.org/10.1093/bioinformatics/btm344.

Savitsky, T., Vannucci, M., Sha, N., 2011. Variable Selection for Nonparametric Gaussian Process Priors: Models and Computational Strategies. Statist. Sci. 26 (1), 130–149. http://dx.doi.org/10.1214/11-STS354.

Schölkopf, B., Smola, A., Müller, K.-R., 1997. Kernel principal component analysis. In: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.-D. (Eds.), Artificial Neural Networks — ICANN'97. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 583–588.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N., 2016. Taking the human out of the loop: A review of Bayesian optimization. Proc. IEEE 104 (1), 148–175. http://dx.doi.org/10.1109/JPROC.2015.2494218.

Shen, H., Yan, Y., Zhao, Z., 2024. DeepDRK: Deep dependency regularized knockoff for feature selection. URL: https://arxiv.org/abs/2402.17176 arXiv:2402.17176.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (56), 1929–1958, URL: http://jmlr.org/papers/v15/srivastava14a.html.

Stoppiglia, H., Dreyfus, G., Dubois, R., Oussar, Y., 2003. Ranking a random feature for variable and feature selection. J. Mach. Learn. Res. 3 (null), 1399–1414.

Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A., 2008. Conditional variable importance for random forests. BMC Bioinformatics 9 (307), 1–11. http://dx.doi.org/10.1186/1471-2105-9-307.

Stuyck, T., Demeester, E., 2024. Impact of using GAN generated synthetic data for the classification of chemical foam in low data availability environments. In: Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - ICPRAM. INSTICC,SciTePress, pp. 620–627. http://dx.doi.org/10.5220/0012305300003654.

Vallerio, M., Navarro-Brull, F.J., 2025. Industrial data github. URL: https://github.com/industrial-data/all-you-need-is-noise.

Vallerio, M., Perez-Galvan, C., Navarro-Brull, F.J., 2024. Industrial data science for batch manufacturing. In: Manenti, F., Reklaitis, G.V. (Eds.), 34th European Symposium on Computer Aided Process Engineering / 15th International Symposium on Process Systems Engineering. In: Computer Aided Chemical Engineering, vol. 53, Elsevier, pp. 2965–2970. http://dx.doi.org/10.1016/B978-0-443-28824-1.50495-6, URL: https://www.sciencedirect.com/science/article/pii/B9780443288241504956.

Viana, F.A.C., 2016. A tutorial on latin hypercube design of experiments. Qual. Reliab. Eng. Int. 32 (5), 1975–1985. http://dx.doi.org/10.1002/qre.1924.

Vitale, R., Westerhuis, J.A., Næ s, T., Smilde, A.K., de Noord, O.E., Ferrer, A., 2017. Selecting the number of factors in principal component analysis by permutation testing—Numerical and practical aspects. J. Chemom. 31 (12), e2937. http://dx.doi.org/10.1002/cem.2937, e2937 cem.2937. URL: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.2937 arXiv:https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.2937.

Weinstein, A., Barber, R., Candes, E., 2017. A power and prediction analysis for knockoffs with lasso statistics. URL: https://arxiv.org/abs/1712.06465 arXiv:1712.06465.

Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometr. Intell. Lab. Syst. 2 (1–3), 37–52. http://dx.doi.org/10.1016/0169-7439(87)80084-9.

Wu, Y., Boos, D.D., Stefanski, L.A., 2007. Controlling variable selection by the addition of pseudovariables. J. Amer. Statist. Assoc. 102 (477), 235–243. http://dx.doi.org/10.1198/016214506000000843.

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

Zuecco, F., Cicciotti, M., Facco, P., Bezzo, F., Barolo, M., 2021. Backstepping methodology to troubleshoot plant-wide batch processes in data-rich industrial environments. Processes 9 (6), http://dx.doi.org/10.3390/pr9061074, URL: https://www.mdpi.com/2227-9717/9/6/1074.