Stream-based active learning with linear models

Davide Cacciarelli [a,b,*], Murat Kulahci [a,c], John Sølve Tyssedal [b]

[a] *Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark*
[b] *Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway*
[c] *Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden*

ARTICLE INFO

ABSTRACT

The proliferation of automated data collection schemes and the advances in sensorics are increasing the amount of data we are able to monitor in real-time. However, given the high annotation costs and the time required by quality inspections, data is often available in an unlabeled form. This is fostering the use of active learning for the development of soft sensors and predictive models. In production, instead of performing random inspections to obtain product information, labels are collected by evaluating the information content of the unlabeled data. Several query strategy frameworks for regression have been proposed in the literature but most of the focus has been dedicated to the static pool-based scenario. In this work, we propose a new strategy for the stream-based scenario, where instances are sequentially offered to the learner, which must instantaneously decide whether to perform the quality check to obtain the label or discard the instance. The approach is inspired by the optimal experimental design theory and the iterative aspect of the decision-making process is tackled by setting a threshold on the informativeness of the unlabeled data points. The proposed approach is evaluated using numerical simulations and the Tennessee Eastman Process simulator. The results confirm that selecting the examples suggested by the proposed algorithm allows for a faster reduction in the prediction error.

1. Introduction

The term big data seems to be ubiquitous in many fields of application, and industrial production is no different. However, in production, this can be somewhat misleading as it often refers to process data that is obtained through automated data collection schemes with minimal manual interference. Product-related data is usually scarcer particularly in high-volume manufacturing due to costs of inspection. This creates an imbalance in the amount of available data that can at times be quite substantial. Yet in many cases, predictive modeling relating process variables to product characteristics is sought after. Therefore, it will be beneficial to guide the data collection schemes for product characteristics through a real-time sampling methodology. In current production environments, sampling of the product characteristics is often performed at regular time intervals or at random. However, this approach can be ineffective as the informativeness of the observations at the time of sampling is not taken into account. This problem is reinforcing the interest of researchers and practitioners in active learning. Active learning-based sampling schemes use an instance selection criterion to strategically select data points that allow a faster reduction of the generalization error [1].

Over the last decades, many active learning approaches have been proposed, but most of the focus has been dedicated to the pool-based scenario [2]. Pool-based active learning refers to a circumstance in which a large amount of unlabeled data is collected all at once and made available to the learner, which can then select offline the data points to be labeled with a greedy approach [3].

In real-time applications for high-volume production, where samples are processed at a fast pace, evaluating all the available instances before making a choice might not be realistic. In these cases, the learner might only have a short time frame to make the sampling decision. Indeed, if a sample is not selected for the quality check, it might get lost in the downstream process and no longer be traceable. This is particularly relevant in high-volume production, where tracing individual parts is a challenge. Also, in a chemical process, we might not be able to measure the level of the variable of interest once a component undergoes a specific treatment. In these contexts, a much more sensible scenario is represented by stream-based active learning, which is sometimes referred to as selective sampling [4]. Stream-based active learning investigates a scenario where instances are processed one at a time and the learner has to determine immediately whether to keep the instance and query its label or discard it. The task is very similar to the one described by a notorious statistical riddle, the secretary problem [5], where an observer sequentially interviews

* Corresponding author at: Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark.
E-mail address: dcac@dtu.dk (D. Cacciarelli).

a certain number of applicants and, after each interview, a decision must be made on whether the applicant is hired or not. An exhaustive survey about stream-based active learning has been proposed by Lughofer [6], who classified existing online active learning methods by taking into account the data processing functionality, the model class (regression or classification), and many other relevant properties. The survey reveals how stream-based active learning methods have been mostly developed in the classification field. Regression models, on the other hand, are extremely useful in the development of soft sensors for hard-to-measure process variables or in quality control problems where a product's characteristic is measured on a continuous scale. That is why active learning in conjunction with regression models is capturing the interest of many researchers [7–10].

In this paper, we focus on the use of linear regression models. These models are well suited for stream-based active learning as they can easily be trained on a small number of observations, being composed of a small number of parameters. This property is also very useful if we want to efficiently retrain the model each time the design is augmented by including an additional observation [11]. Moreover, despite recent advances in terms of interpretability for deep learning models, linear regression models are still amongst the most easily interpretable models. Indeed, their parameters offer a straightforward quantitative contribution of each specific feature, and their input features are directly derived from the empirical observations [12]. Besides the direct interpretation that comes from the signs and magnitudes of the coefficients, linear models can also be used to construct confidence intervals on the parameter estimates and variable selection can also be easily incorporated into such models [13]. Recently, additional variable selection methods for linear regression models have been suggested by Zhang et al. [14]. Being able to offer a robust feature importance analysis is particularly important in industrial problems, where practitioners and engineers might need to timely intervene in specific parts or components of the process to ensure safety and operational efficiency. The simplicity of these models and the low number of parameters that require tuning is also beneficial to foster their adoption and use in applications. Finally, linear regression models allow us to build on the optimal experimental design theory and leverage the criteria that are typically used to design highly efficient experiments. Despite the focus of this paper being dedicated to linear models, nonlinear models proved to be extremely useful in a wide variety of applications. In particular, deep learning models are very effective in dealing with complex high-dimensional data to perform tasks like image recognition, shape extraction, and pose recovery [15–19].

In this work, we propose a novel strategy to perform stream-based active learning with linear models. Given the impossibility to rank observations in real-time, we provide an algorithm that only uses unlabeled data to set a threshold on the informativeness of data points. Unlabeled data is also exploited in a semi-supervised manner to increase the predictive performance [20]. We show how the proposed approach outperforms random sampling and state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2, we define some basic concepts and discuss related works focusing on active learning for regression. Section 3 introduces the proposed sampling strategy. In Section 4 we test our approach using numerical simulations; the Tennessee Eastman Process data is also used to evaluate its performance on a typical industrial process. Finally, Section 5 provides some conclusions.

2. Preliminaries

The active learning problem is defined by an imbalance between the availability of process variables $\mathbf{x} \in \mathbb{R}^p$ and the

corresponding labels $y \in \mathbb{R}$. In many circumstances, industrial processes are characterized by the presence of easy-to-measure process variables, which are collected through automated collection schemes, and hard-to-measure variables, whose values are difficult to track during routine operations. Large plants, measurement delays, and environments hostile to the survival of measuring devices are all situations where hard-to-measure process variables are commonly encountered [21]. Similar situations can be addressed by utilizing soft sensors based on predictive models to forecast the true values of hard-to-measure variables. For modeling purposes, we assume that the true underlying relationship between the process variables and the product information or hard-to-measure variable can be expressed with a linear model of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$\mathbf{y}$ is a $n \times 1$ vector of response variables, $\mathbf{X}$ is a $n \times p$ model matrix, $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients, and $\boldsymbol{\epsilon}$ is a $n \times 1$ vector representing the noise, with covariance matrix $\sigma^2\mathbf{I}$. Here n represents the total number of observations and p the number of process variables (as well as the number of parameters in a model with main effects only and no intercept). If the predictors and the response are not centered, an intercept term may be added to the model. In that case, the size of the model matrix becomes $n \times (p+1)$, and $\boldsymbol{\beta}$ a $(p+1) \times 1$ vector. When $k \geq p$ observations are available to the learner, we can obtain a least squares estimator for $\boldsymbol{\beta}$ using

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y} \tag{2}$$

such that the fitted linear regression model will be given by $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ and its residuals by $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$. A key distinction between the experimental design approach and stream-based active learning concerns the assumption we make about the randomness of the process variables. In design of experiments, the $\mathbf{x}$ vectors are assumed to be fixed while in this case we assume that $\mathbf{X}$ is composed by random vectors, as the individual observations are sampled from a process subject to random variation and we are not able to set the precise location of the incoming data points. However, conditional on the observed $\mathbf{X}$ variables, $(\mathbf{x}_1, \ldots, \mathbf{x}_p)$, Eq. (2) still applies. It should be noted that the coefficients $\hat{\boldsymbol{\beta}}$ determined using Eq. (2) may not be stable if the data matrix $\mathbf{X}$ is affected by multicollinearity. To deal with this issue and achieve robust results, a solution might be to use a ridge estimate for the coefficients, $\hat{\boldsymbol{\beta}}_{\text{ridge}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} - \lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}$. An alternative approach to tackle multicollinearity is to perform a pre-whitening of $\mathbf{X}$ to remove the dependencies between the components.

We assume a small, labeled training set is initially available and can be used to fit the first regression model, as is common practice in active learning applications [22–24]. The number of observations provided to the learner usually corresponds to a modest fraction (e.g., 5%) of the total number of

instances available [25,26]. After the first model has been built, the learner is granted a certain operational budget b to augment the design matrix by including additional observations. Some approaches focus on this problem in a pool-based context, in which the total number of observations n is represented by a closed and static set $\mathcal{U}$ and the label of a specific data point can always be queried. Among these approaches, query-by-committee (QBC) [22] suggests building an ensemble of regression models trained on bootstrap replica of the original training set. Once the ensemble, or committee, has been built, the variance of the predictions made by the committee members is computed for each unlabeled observation $\mathbf{x} \in \mathcal{U}$. This metric, also referred to as ambiguity, is used to rank the instances belonging to the unlabeled set $\mathcal{U}$ by prioritizing the data points with the highest variance. Expected model change maximization (EMCM) [26] is another noteworthy study that focuses on the observations that impact the most the current model's parameters. The model change is defined as the difference between the current model parameters and the parameters obtained after fitting the model on the augmented design, including the unlabeled observation $\mathbf{x} \in \mathcal{U}$ that is currently under evaluation. Because the learner does not have access to the true label for that data point, it estimates it using the mean prediction of a bootstrap ensemble, as the one employed by QBC. Another offline approach, inspired by statistical process control, combines the Hotelling T^2 statistic and the squared prediction error of a principal component regression (PCR) model to obtain a sampling evaluation index [23].

Besides the fact that all these methods focus on the pool-based scenario, it should be noted that the approaches that use ensembles may not be well suited for the online scenario, given the higher computational cost associated with training and updating the models.

Optimal experimental theory is another field of research that is intrinsically related to active learning [27,28]. Optimal designs aim to reduce the cost of experimentation by proposing design matrices that allow a robust parameter estimation with the minimum number of runs. The most commonly employed optimality criteria are D-optimality [29] and A-optimality. Important properties of a design can be derived from the moment matrix, or information matrix, which is defined as

$$\mathbf{M} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{N} \tag{3}$$

where N represents the total number of runs. The moment matrix specifies the distribution of points in space and can be used to describe the design geometry. In a 2^k factorial design, where variables are expressed in coded units $(-1, +1)$, the moment matrix is equal to the identity matrix $\mathbf{I}_k$, as the columns of the design are orthogonal. In an orthogonal design, all the parameters can be estimated independently of one another [30]. D-optimal designs try to pursue such property by focusing on good model parameter estimation. Inverting the moment matrix we obtain the scaled dispersion matrix given by

$$\mathbf{M}^{-1} = N\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \tag{4}$$

This matrix contains the variances and covariances of the estimated coefficients of the regression model, scaled by N/σ^2 [28]. Indeed, if the k observations used to estimate $\hat{\boldsymbol{\beta}}$ are i.i.d. and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, we have

$$\hat{\boldsymbol{\beta}}_k|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{\beta}, \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\boldsymbol{\Sigma}^2\right) \tag{5}$$

It can be demonstrated how by increasing the determinant of $\mathbf{M}$, the variances and covariances of the model coefficients are reduced, leading to a better estimation of $\boldsymbol{\beta}$. A D-optimal design is

attained by maximizing the determinant of the moment matrix. Formally, we are seeking the design $\mathcal{D}^*$ that satisfies

$$\max_{\mathcal{D}} |\mathbf{M}(\mathcal{D})| = |\mathbf{M}(\mathcal{D}^*)| \tag{6}$$

A-optimality is another important optimality criterion that tries to achieve good parameter estimation by minimizing the sum of the individual variances of the coefficients. This is achieved by the design $\mathcal{D}^*$ that satisfies

$$\min_{\mathcal{D}} \mathrm{tr}\left[\mathbf{M}(\mathcal{D})\right]^{-1} = \mathrm{tr}\left[\mathbf{M}(\mathcal{D}^*)\right]^{-1} \tag{7}$$

as the variances of the coefficients can be found on the diagonal of the scaled dispersion matrix multiplied by σ^2/N. It should be noted that A-optimality does not consider the covariances between coefficients.

Recently, the concept of A-optimality has been extended to stream-based active learning [31,32]. That is, the approach has been extended outside the design of experiments framework, assuming $\mathbf{X}$ is composed of random vectors and the observations are sequentially drawn. Riquelme et al. [32] show how to set a threshold to perform online active learning for linear regression models by minimizing the sum of the individual variances of $\hat{\boldsymbol{\beta}}$. They state that, in order to achieve A-optimality and minimize the trace of the inversed scaled dispersion matrix, the eigenvalues of the moment matrix should be as balanced as possible. This is because the eigenvalues of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ represent the trace of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, which is also given by the sum of the norm of the observations. For this reason, they propose a norm-thresholding algorithm that pursues A-optimality by selecting observations with large, scaled norm. The scaling step can be ignored when whitening is used to remove dependencies. Finally, the design is augmented with the observations $\mathbf{x}$ whose norm exceeds a threshold Γ given by

$$\mathrm{P}_{\mathrm{D}}\left(\|\mathbf{x}\| \geq \Gamma\right) = \alpha \tag{8}$$

where α is the ratio of observations we are willing to label out of the incoming data stream. This value is strongly dependent on the budget b and the sampling rate used to collect the data.

Another noteworthy approach focusing on stream-based active learning for regression tasks has been suggested by Lughofer and Pratama [33]. In this paper, the authors propose a single-pass selection criterion that takes into account ignorance about the input space, uncertainty in predictive model outputs, and uncertainty in model parameters. The main difference with our approach is that Lughofer and Pratama focus on the use of Takagi–Sugeno (TS) fuzzy models [34], combining adaptive error bars for the model output and A-optimality for the variances of the estimated parameters. Conversely, our method relies on statistical linear regression and tries to combine the exploration of lesser-known input space regions with accurate parameter estimates by employing the idea of D-optimality.

3. Proposed approach

In this work, we try to improve the approach proposed by Riquelme et al. [32] by moving from A-optimality to D-optimality. We believe that taking into account the covariance between the estimates of the model coefficients might be particularly advantageous with large datasets and models, where many factors might be active and influence the response. To adapt the D-optimality criterion to stream-based active learning, we start from the connection between D-optimality and prediction variance (PV) highlighted by Myers et al. [28]. The PV at a point $\mathbf{x}^{(m)}$ is the variance of the predictor $\hat{\mathbf{y}}(\mathbf{x}^{(m)})$, which corresponds to $\mathrm{Var}(\mathbf{x}^{(m)\mathrm{T}}\hat{\boldsymbol{\beta}})$, and is given by

$$\mathrm{PV}(\mathbf{x}) = \sigma^2 \mathbf{x}^{(m)\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}^{(m)} \tag{9}$$

where $\mathbf{x}^{(m)}$ represents the data point where the variance is being estimated, expanded to the model form. We can also express the variance in a scale-free form using the scaled prediction variance (SPV), which is computed as

$$\text{SPV}(\mathbf{x}) = N\mathbf{x}^{(m)\text{T}}\left(\mathbf{X}^\text{T}\mathbf{X}\right)^{-1}\mathbf{x}^{(m)} \tag{10}$$

It should be noted that the SPV is a quadratic form of the inverse moment matrix $\mathbf{M}^{-1}$, as it can also be written as $\mathbf{x}^{(m)\text{T}}\mathbf{M}^{-1}\mathbf{x}^{(m)}$. Since SPV considers the total number of runs N, it can be used to assess the quality of a design on a per observation basis. In the online scenario, we are not interested in comparing designs of different sizes but rather we investigate the individual contributions of incoming data points to the current design. In this circumstance, we can discard N and use the unscaled prediction variance (UPV), which is calculated as

$$\text{UPV}(\mathbf{x}) = \mathbf{x}^{(m)\text{T}}\left(\mathbf{X}^\text{T}\mathbf{X}\right)^{-1}\mathbf{x}^{(m)} \tag{11}$$

As anticipated in Section 2, we are already given an initial random design that contains some labeled examples, which is being used to fit an initial model. Then, we are interested in augmenting our design by iteratively selecting observations from a continuous stream. Pursuing D-optimality, we aim at collecting observations that allow us to maximize the determinant of the moment matrix $\mathbf{M}$. If we consider that the current design is composed by k observations, we can decompose the numerator of the moment matrix (Eq. (3)) before the design is augmented by including the $(k+1)th$ observation as

$$\mathbf{X}_k^\text{T}\mathbf{X}_k = \mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^\text{T} \tag{12}$$

we can then express the determinant of $\mathbf{X}_k^\text{T}\mathbf{X}_k$ as the product of the determinant of the numerator of the augmented moment matrix and a second term as in

$$\begin{aligned}|\mathbf{X}_k^\text{T}\mathbf{X}_k| &= |\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^\text{T}| \\ &= |\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1}||1 - \mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}|\end{aligned} \tag{13}$$

It should be noted that the second term of the above equation is a scalar, irrespective of the number of variables p and the number of observations k. From there, we can observe that

$$\frac{|\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1}|}{|\mathbf{X}_k^\text{T}\mathbf{X}_k|} = \frac{1}{1 - \mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}} \tag{14}$$

From the properties of the hat matrix, which is generally defined as $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^\text{T}\mathbf{X}\right)^{-1}\mathbf{X}^\text{T}$, we know that $0 \leq h_{jj} \leq 1$ is true for each element h_{jj} of $\mathbf{H}$ [35]. It follows that $\mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1} \leq 1$. Hence, we can conclude that the determinant of the new, enlarged, training set is maximized by seeking observations $\mathbf{x}$ that maximize $\mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}$. That is, we will only select points that maximize the UPV. This may be explained by the fact that a data point for which we have a large prediction variance represents a less known region of the input space, and the regression model will highly benefit from its inclusion in the design. From Myers et al. [28] we have that maximizing $\mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_{k+1}^\text{T}\mathbf{X}_{k+1}\right)^{-1}\mathbf{x}_{k+1}$ is equivalent to maximizing $\mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_k^\text{T}\mathbf{X}_k\right)^{-1}\mathbf{x}_{k+1}$, which is the UPV using the fitted model before the new point has been added to the training set.

Finally, following the norm-thresholding approach, we can set an upper control limit on new observations as

$$\text{P}_\text{D}\left(\mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_k^\text{T}\mathbf{X}_k\right)^{-1}\mathbf{x}_{k+1} \geq \Gamma\right) = \alpha \tag{15}$$

In practice, as suggested by Riquelme et al. [32], when we start to observe the data points coming from the process, we allocate a first initial set of points to estimate the distribution

of $\mathbf{x}_{k+1}^\text{T}\left(\mathbf{X}_k^\text{T}\mathbf{X}_k\right)^{-1}\mathbf{x}_{k+1}$. In this work, we used kernel density estimation (KDE) with a Gaussian kernel. The initial set is also used to estimate the sample covariance matrix Σ. By performing an eigenvalue decomposition we can then express Σ as $\mathbf{U}\Lambda\mathbf{U}^\text{T}$, where $\mathbf{U}$ is an orthogonal matrix, whose ith column corresponds to the ith eigenvector of Σ, and Λ is a diagonal matrix with the eigenvalues of Σ on the diagonal. The incoming observations $\mathbf{x}$ can then be whitened using

$$\mathbf{z} = \Lambda^{-1/2}\mathbf{U}^\text{T}\mathbf{x} \tag{16}$$

Before the whitening step, data can be centered and scaled using the sample mean and variances obtained from the initial set. In industrial contexts, when a lot of unlabeled process data is available in the form of a historical database, this step can also be performed offline. In this case, by fitting a principal component analysis (PCA) model to the large unlabeled dataset and using it to transform the incoming observations, we could improve the predictive performance using a semi-supervised PCR as suggested by Frumosu and Kulahci [20]. The use of semi-supervised classification models has also received some attention in active-learning problems [36–38]. Indeed, semi-supervised learning and active learning are both techniques that deal with scarcity of labels. However, they do so in two different ways. With semi-supervised learning, we try to get the most out of the currently available unlabeled data, whereas with active learning we try to acquire new data in the most effective way.

Algorithm 1 describes the complete stream-based active learning procedure with the proposed approach, which might also be referred to as conditional D-optimality (CDO).

An alternative representation of the CDO active learning routine is reported in the flowcharts in Figs. 1 and 2. The first flowchart depicts the warm-up phase, which is represented by the steps from 1 to 10 of Algorithm 1. The warm-up set is very important for the algorithm and serves two main purposes. First, it allows to estimate the covariance matrix of the data, which is later used for whitening the incoming observations. Secondly, it provides a set of unlabeled observations that can be leveraged to estimate the distribution of the UPV. The primary purpose of the whitening step is to address the multicollinearity issue in linear regression modeling, which can be aggravated when dealing with real-world data. The whitening step also ensures comparability with the norm-thresholding approach. Indeed, the norm-thresholding method without whitening would require computing a weighted norm to deal with dependencies between the components.

The second flowchart represents the instance selection phase, the core of the active learning strategy. At this stage, we compute the UPV for the new observation sampled from the stream and we compare it to a pre-defined threshold. If the UPV computed at this point exceeds Γ, we query its label and include the labeled example in the training set. After the inclusion of the new point, a new threshold is estimated. The threshold is found by applying Eq. (15) to the whitened warm-up set $\mathbf{V}$. That is, $\mathbf{X}_k$ is substituted by $\mathbf{Z}$, the currently labeled training set after whitening, and $\mathbf{x}_{k+1}$ is given by each unlabeled data point belonging to $\mathbf{V}$. By doing so, we obtain a one-dimensional array that has the same cardinality as the number of observations in $\mathbf{V}$. These statistics are then used to approximate the distribution of the UPV using KDE and determine the α-upper percentile.

4. Experiments

In the experiments, we compare the proposed method to the norm-thresholding approach and random sampling. The methods are tested using numerical simulations and data from a chemical process simulator. All the approaches start from the same labeled

---

Algorithm 1 Stream-based active learning using CDO

Input: an initial random design $\mathbf{X}$; a data stream $\mathcal{S}$; a warm-up length w; a sampling rate α; a budget b
Output: an augmented design $\mathbf{Z}$

1:    Set $\mathbf{W} = \emptyset$ ▷ storing initial data to estimate $\mathbf{\Sigma}$ and Γ
2:    $i \leftarrow 1, c \leftarrow 0$ ▷ c represents the currently used budget
3:    **while** $i \leq w$ **do**
4:        Observe the ith data point $\mathbf{x}_i \in \mathcal{S}$
5:        Select $\mathbf{x}_i$: $\mathbf{W} = \mathbf{W} \cup \mathbf{x}_i$
6:        $i \leftarrow i + 1$
7:    **end while**
8:    Estimate the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{W}$ and perform eigendecomposition $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}$
9:    Whiten the initial design by computing $\mathbf{Z} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^{\mathrm{T}} \mathbf{X}$
10:   Whiten the warm-up observations by computing $\mathbf{V} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^{\mathrm{T}} \mathbf{W}$
11:   Estimate Γ using KDE on $\mathbf{V}$ with the desired sampling rate α using Equation 15 and datasets $\mathbf{Z}$ and $\mathbf{V}$
12:   **while** $c \leq b$ **and** $i \leq |\mathcal{S}|$ **do**
13:       Observe the ith data point $\mathbf{x}_i \in \mathcal{S}$
14:       Whiten $\mathbf{x}_i$ by computing $\mathbf{z}_i = \mathbf{\Lambda}^{-1/2} \mathbf{U}^{\mathrm{T}} \mathbf{x}_i$
15:       **if** $\mathbf{z}_i^{\mathrm{T}}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}\mathbf{z}_i \geq \Gamma$ **then**
16:           Ask for the label y_i and augment the labeled dataset $\mathbf{Z} = \mathbf{Z} \cup \mathbf{z}_i$
17:           $c \leftarrow c + 1$
18:           Estimate new threshold Γ to measure the prediction variance of the enlarged design (step 11)
19:       **else**
20:           Discard $\mathbf{x}_i$
21:       $i \leftarrow i + 1$
22:       **end if**
23:    **end while**
24:    **return** $\mathbf{Z}$

---

training set and then they iteratively augment the design until the budget constraint b is met. The performance of the models is expressed, in predictive terms, by the root mean squared error (RMSE) of the predictions on a separate test set of n observations

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{17}$$

4.1. Numerical simulations

To analyze the validity of the proposed method in the stream-based scenario, multiple datasets were created, each with a different dimensionality in terms of the number of process variables p. Within each dataset, incoming observations $\mathbf{x}$ are distributed according to a joint multivariate normal distribution $\mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}_0)$, where $\mathbf{\Sigma}_0$ is given by $\sigma^2 \mathbf{I}$, with $\sigma^2 = 1$. We ran 50 simulations for each number of p and, for each simulation run, the true coefficients are generated as $\boldsymbol{\beta} \sim U(-5, 5)$. It should be noted that $\boldsymbol{\beta}$ has the same dimensionality as $\mathbf{x}$. This means that, using a first order model, a coefficient for each process variable needs to be estimated. The noise is given by $\epsilon \sim \mathcal{N}(0, 1)$. For each scenario, an initial random design $\mathbf{X}$ is assumed available to the learner. We selected $p + 2$ number of observations for the initial design, as $k \geq p$ observations are needed to uniquely estimate $\hat{\boldsymbol{\beta}}$.

The learning curves reported in Figs. 3 and 4 show the difference between the RMSE obtained with the two active learning strategies, using random sampling as the baseline. For each learning step, the percentage RMSE difference reported in the plots is obtained by computing $(\mathrm{RMSE}_{\text{Active Learning}} - \mathrm{RMSE}_{\text{Random}})/$ $\mathrm{RMSE}_{\text{Random}}*100$. This allows us to display a scale-free performance metric while comparing the different scenarios. The plots reporting the learning curves with the absolute RMSE values are included in the appendix.

The methods are tested using $b = 50$ and with different levels for the α shown in Eq. (8), and 15. In the case of random sampling, α represents the probability of selecting an incoming observation. That is, each time a new sample arrives, a number $s \sim U(0, 1)$ is generated and the data point is only selected if

$s \geq 1 - \alpha$. The warm-up length w was set to 500 observations and it is being used by all the methods to estimate the covariance matrix, which is used for whitening the observations in a semi-supervised fashion. Moreover, it ensures comparability between the three strategies by setting the same starting points for the data streams. The models have been fitted without the intercept term as both process variables and outcome are centered.

Fig. 3 shows the performance when using an α equal to 10%. The x-axis reports the learning steps, which correspond to the inclusion of an additional observation to the training set. Indeed, when the design is augmented, the model is updated and new predictions are obtained for the same separate test set. It should be noted that the RMSE obtained in the first learning step is the same for the three methods, as all the models start from the same random design. It can be seen how the performance of the two active learning methods converges to the one obtained through random sampling as the number of labeled examples in the training set increases. Instead, when the number of labeled examples is lower, active learning proves to be particularly convenient. However, the proposed approach dominates the other strategies in all the scenarios. Furthermore, it should be noted how the norm-thresholding algorithm seems to worsen when more and more parameters need to be estimated. Instead, CDO consistently provides enhanced predictive performances. We believe this may be due to the fact that, by imposing a threshold on the norm, A-optimality seeks only points that are far from the design's center, without ensuring a distance between the data points that have already been collected. CDO, on the other hand, emphasizes points that correspond to a poor prediction, which is more likely associated with a design area that the learner has not thoroughly explored. As a result, we are less prone to acquire the labels of data points in locations where we have already collected a significant number of observations.

It should be noted that in real-time applications the improvement offered by active learning is not as large as the one that can be obtained in offline scenarios, where we can deterministically maximize the desired optimality criterion over a closed set of observations. Moreover, by setting $\alpha = 10\%$ we are not being too demanding in terms of selecting observations with large norms for the A-optimality or high prediction variances for CDO. In Fig. 4, we try to widen the gap with the random strategy by lowering α, in this case up to 0.01. By raising the threshold, we can be more
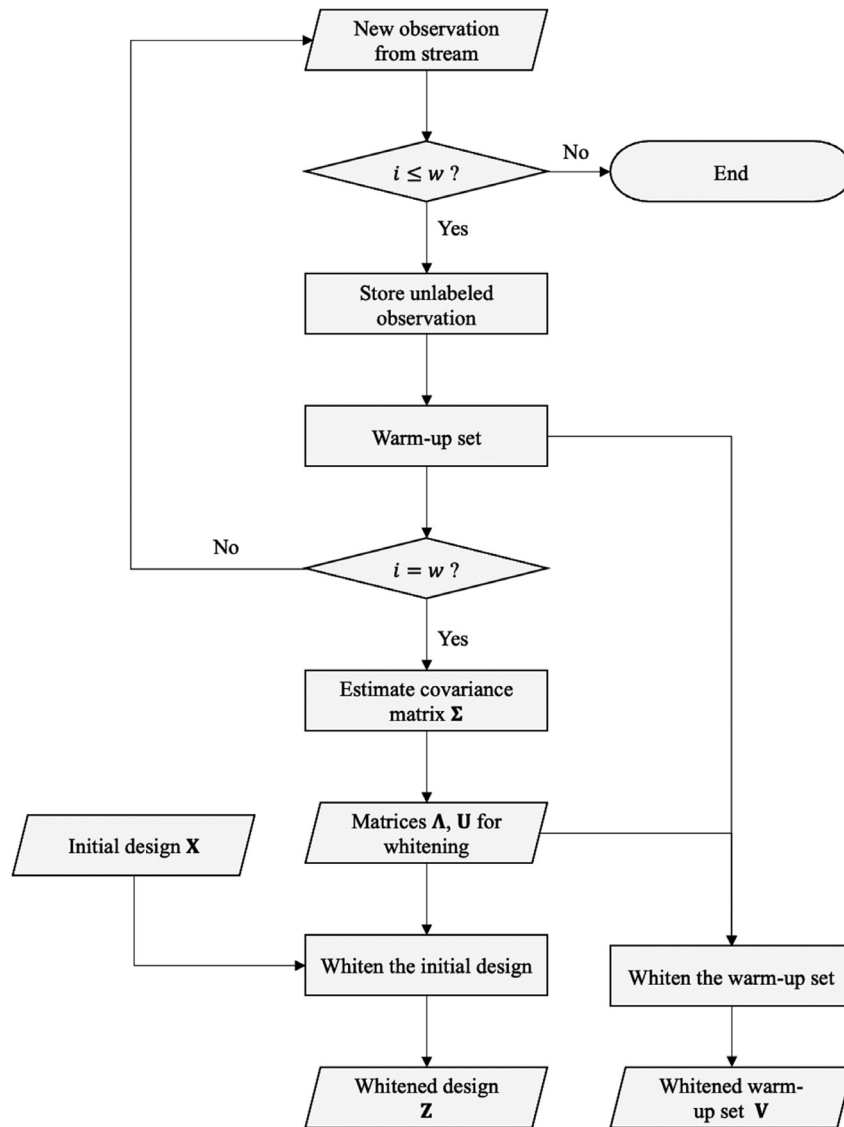
Fig. 1. Flowchart of the warm-up phase of the stream-based active learning procedure.

demanding in terms of the desirability of the selected instances. The only drawback is that the algorithms will need to span more observations to achieve the desired size for the augmented design and meet the budget constraint. We believe this may not represent an issue since data is nowadays collected at very high sampling rates. However, in the final decision concerning the level of α, practitioners will need to make a trade-off between the desired prediction improvement and the time required to select the new labeled examples.

Fig. 4 reports the learning curves obtained using a smaller α. As expected, the enhancement obtained using the proposed strategy is increased with respect to the passive random sampling. However, it is worth noting that the improvement is more evident when the number of parameters is smaller, as the gain obtained in the high-dimensional cases was already significant with $\alpha = 10\%$.

Finally, we analyze the computational time required by the two active learning strategies. To this extent, we introduce a measure called average decision time, which quantifies the time required to decide whether to query the label of an unlabeled observation or discard it. The results obtained on the numerical simulations, for different number of process variables, are

Table 1
Average decision times (ms) for the two active learning methods (50 variables).

Strategy	10 variables	20 variables	50 variables	100 variables
CDO	0.00494	0.00527	0.00568	0.00690
Norm-thresholding	0.00635	0.00642	0.00673	0.00716

reported in Table 1. Both active learning strategies are highly efficient and do not require a high computational time. According to the CDO strategy, at each iteration we are simply computing the UPV for the new data point, as in step 15 of Algorithm 1, which requires less time than computing the norm of the new observation. It should be noted that the average decision time is lower because the inverse of the whitened moment matrix, $\left(\mathbf{Z}^{\mathsf{T}}\mathbf{Z}\right)^{-1}$, does not need to be computed at each iteration. However, it must be updated when the design is augmented by including an additional labeled observation. Updating and inverting the whitened moment matrix takes, on average, 0.31375 ms (ms).

From an operational point of view, the average decision time is a highly relevant metric and it is closely related to the specific
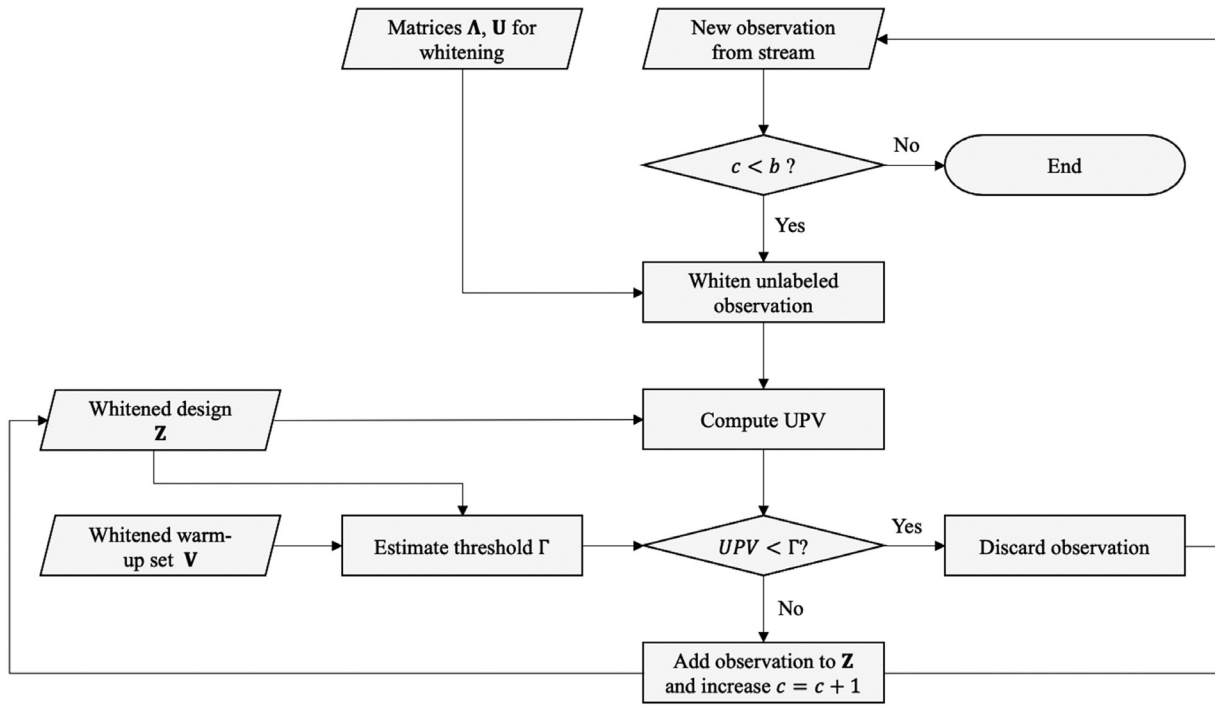
Fig. 2. Flowchart of the instance selection phase of the stream-based active learning procedure.
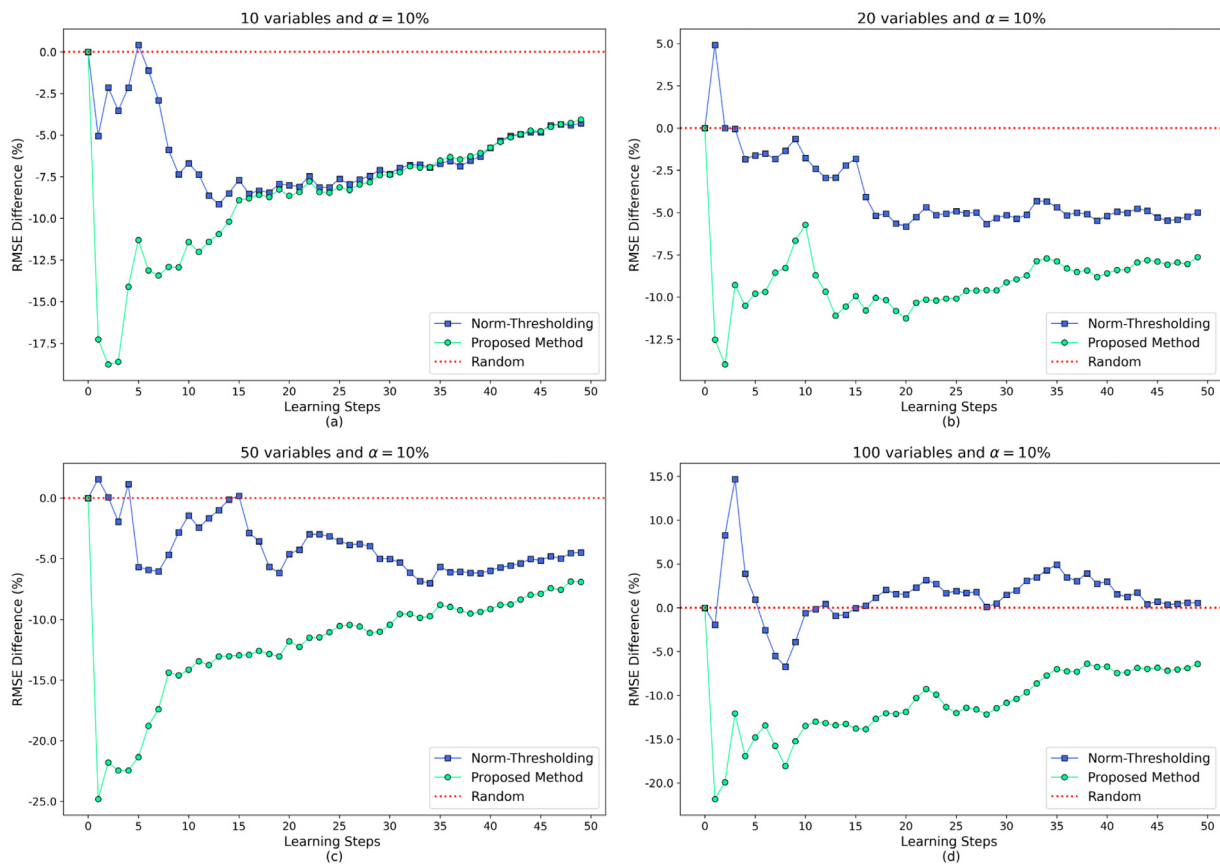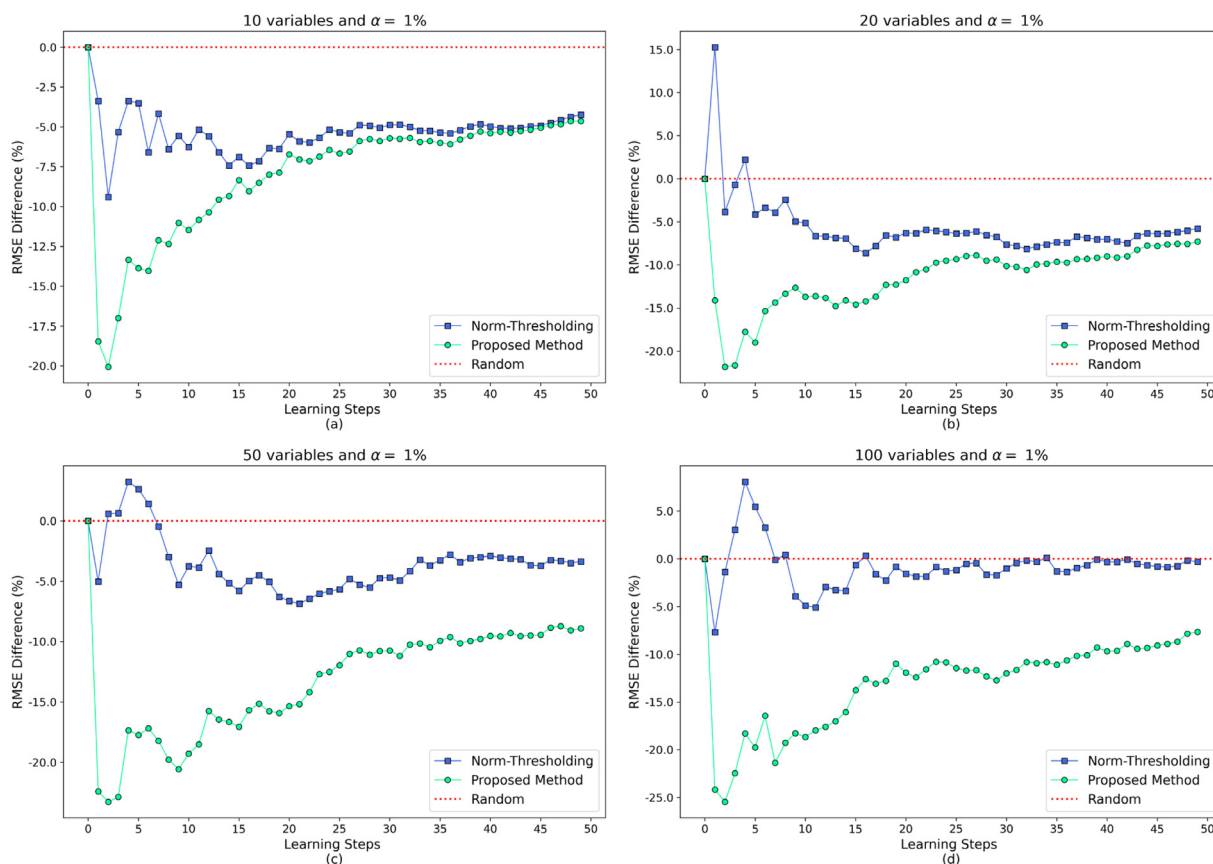

Fig. 3. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 10\%$ (50 simulations).

sampling frequency of the process. Indeed, to allow for a timely instance selection, the decision time should be strictly lower than the expiry date of the unlabeled data point, which is given by the time window where it is possible to query its label.

4.2. Tennessee Eastman Process

The Tennessee Eastman Process (TEP) is a commonly used benchmark in industrial and chemical engineering research and

Fig. 4. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 1\%$ (50 simulations).
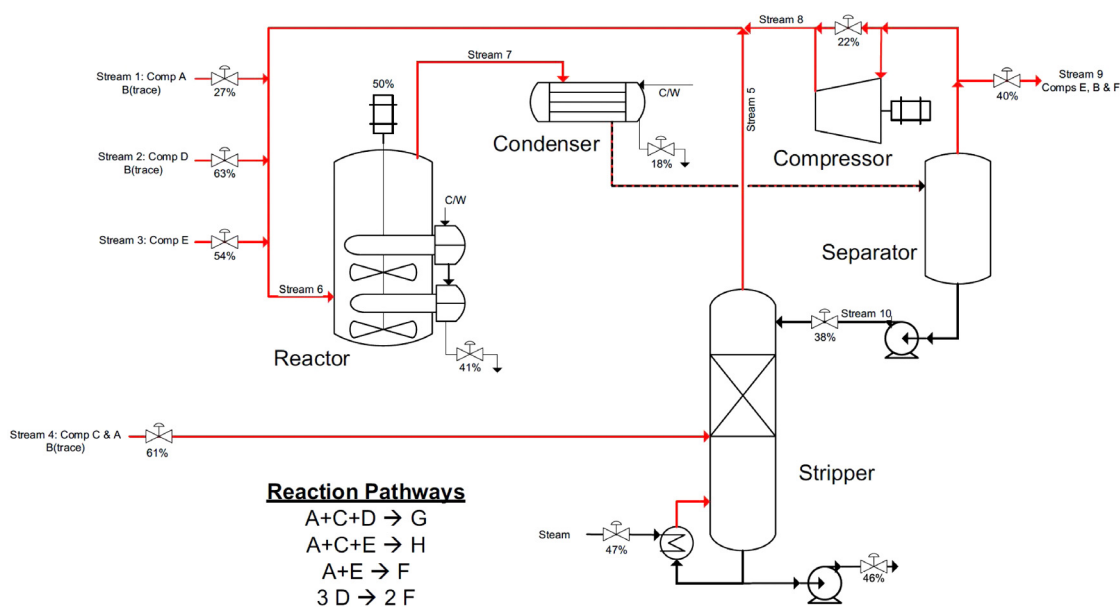
Fig. 5. The TEP piping and instrumentation diagram [39].

it has been thoroughly investigated in terms of process dynamics and control [40–44]. Recently, it has been also used to validate active learning or soft sensor modeling approaches [45–49]. It was initially published in 1993 [50] but since then it has been further developed and improved. For this study, we used a recently released MATLAB simulator to generate the data [39,51]. We generated 50 datasets with the process running in normal operating conditions, using a sampling rate of approximately 1 min. Fig. 5 depicts the TEP flowchart, which shows how the process is primarily composed of a reactor, a product condenser and separator, a stripper, and a compressor.

The TEP, like many other industrial processes, includes some easy-to-measure process variables whose real value can easily be monitored online, and some hard-to-measure variables,

Table 2
Variables of the TEP used as predictors in the regression models.

Number	Process variable	Code	Number	Process variable	Code
1	A feed	XMEAS1	9	Product separator temperature	XMEAS11
2	D feed	XMEAS2	10	Product separator pressure	XMEAS13
3	E feed	XMEAS3	11	Product separator underflow	XMEAS14
4	A and C feed	XMEAS4	12	Stripper pressure	XMEAS16
5	Recycle flow	XMEAS5	13	Stripper temperature	XMEAS18
6	Reactor feed rate	XMEAS6	14	Separator steam flow	XMEAS19
7	Reactor temperature	XMEAS9	15	Reactor cooling water outlet temperature	XMEAS21
8	Purge rate	XMEAS10	16	Separator cooling water outlet temperature	XMEAS22
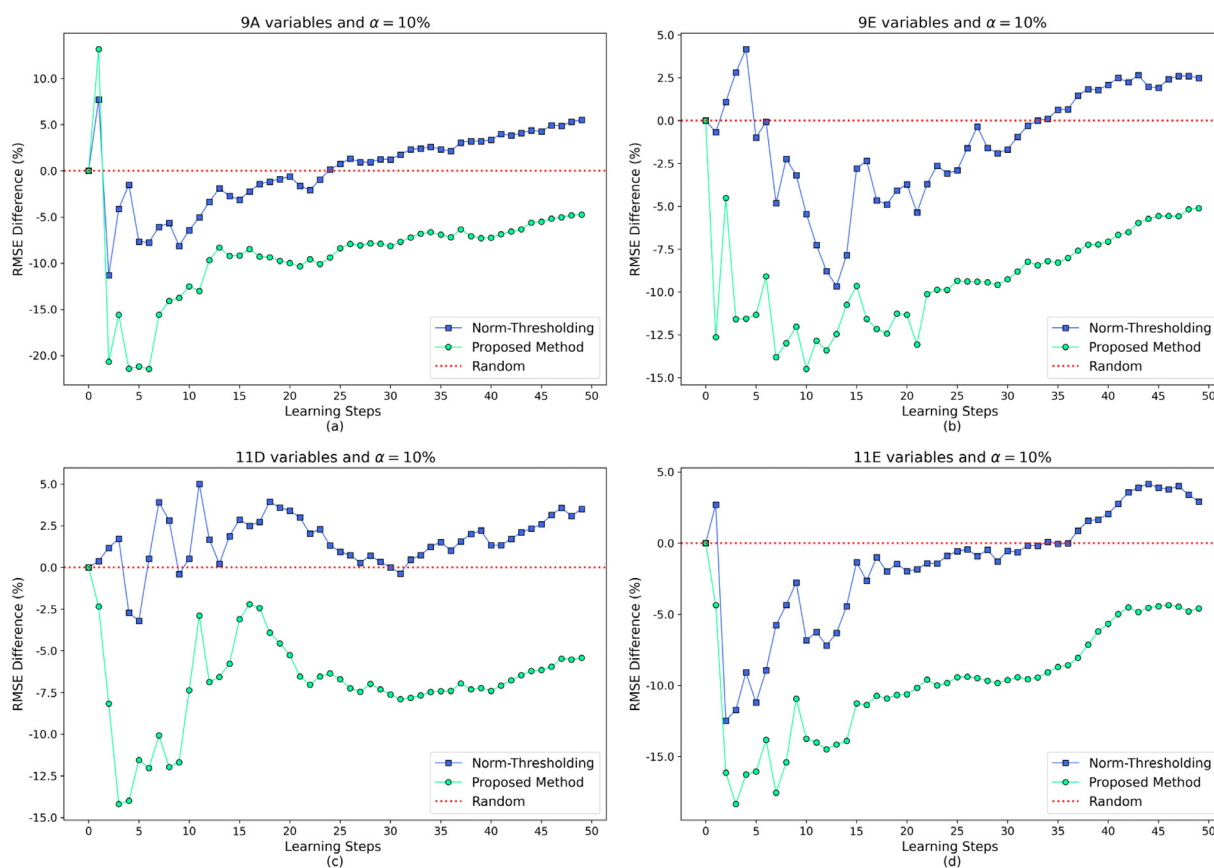
Fig. 6. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 10\%$ (50 simulations).

which are difficult to track during routine operations. Data-driven soft sensors are often developed to predict the latter in real-time. However, training regression models frequently necessitates a large number of labeled examples, and conducting quality inspections on chemical products may be costly and time-consuming. For this reason, optimizing the sampling strategy using active learning is highly desirable.

The 16 process variables shown in Table 2 are often used as predictors for the hard-to-measure process variables when testing active learning or soft sensor modeling approaches on the TEP. In most cases, the response variable is one of the composition measurements, such as the purge or product streams [45,47,48]. In this work, we selected two purge streams (Stream 9 A and Stream 9E) and two product streams (Stream 11D and Stream 11E) as the response to be predicted using the easy-to-measure variables.

As in the case of the numerical simulations, 50 datasets have been generated, and the average RMSE results are presented in the learning curve plots in Figs. 6 and 7. Most of the experimental parameters correspond to the ones used in the numerical study. The number of observations allocated to the first training set is

equal to $p+2$, which in this case corresponds to 18. The warm-up length w is equal to 500 and the budget b is set to 50. The main difference from the models used in Section 4.1 is that, in this case, all the models include the intercept term.

We can see in Fig. 6 how the results obtained in Section 4.1 are still valid with data coming from a realistic industrial process simulator. Indeed, both the random and norm-thresholding approaches are outperformed by the proposed strategy. With regards to the level of α, the behavior observed in the numerical study does not seem to be altered and, as the threshold is raised, the performance gap between random sampling and active learning strategies widens. The plots of the learning curves with the absolute RMSE values are included in the appendix.

The plots in Fig. 8 show the residuals related to the first composition measurements analyzed, stream A of the purge. For illustrative purposes, the residuals refer to a smaller test set, composed of 100 observations. The first plot (a) shows the residuals obtained with the first random design, which is common to all the compared approaches. The remaining plots (b–d) illustrate the residuals obtained after five learning steps with each strategy. In general, we can see how the predictive performance improves
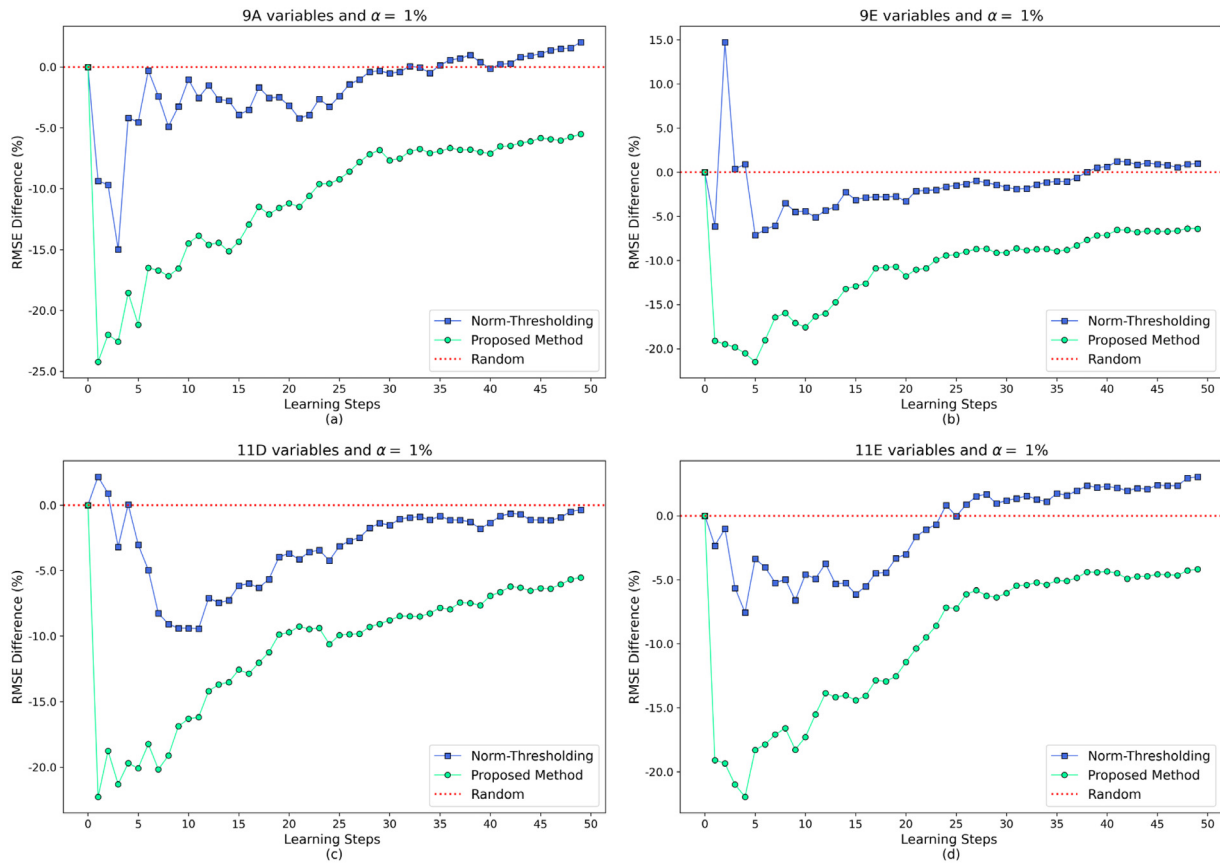
Fig. 7. Percentage difference in RMSE between random sampling and the active learning methods, using $\alpha = 1\%$ (50 simulations).
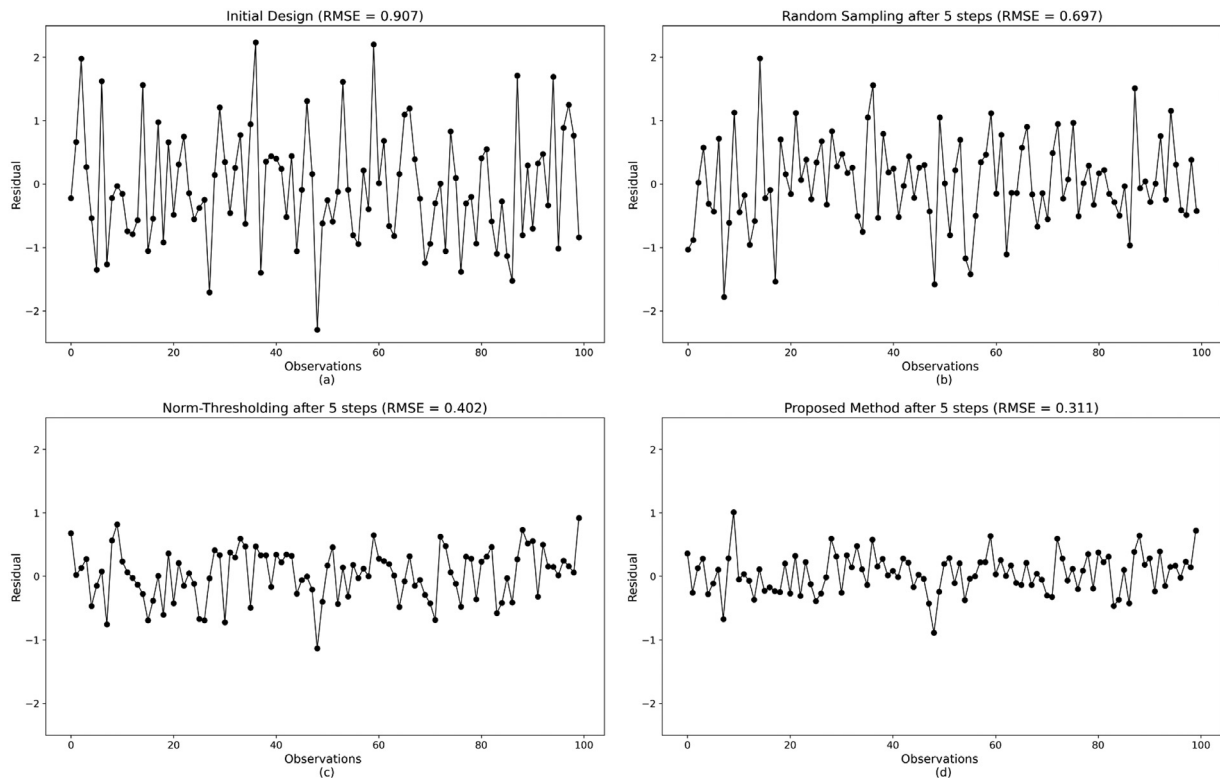

Fig. 8. Residuals of the Stream 9 A predictions: with the initial training set (a) and after augmenting the design with 5 additional labeled examples with the different methods (b–d) (one simulation with $\alpha = 1\%$).
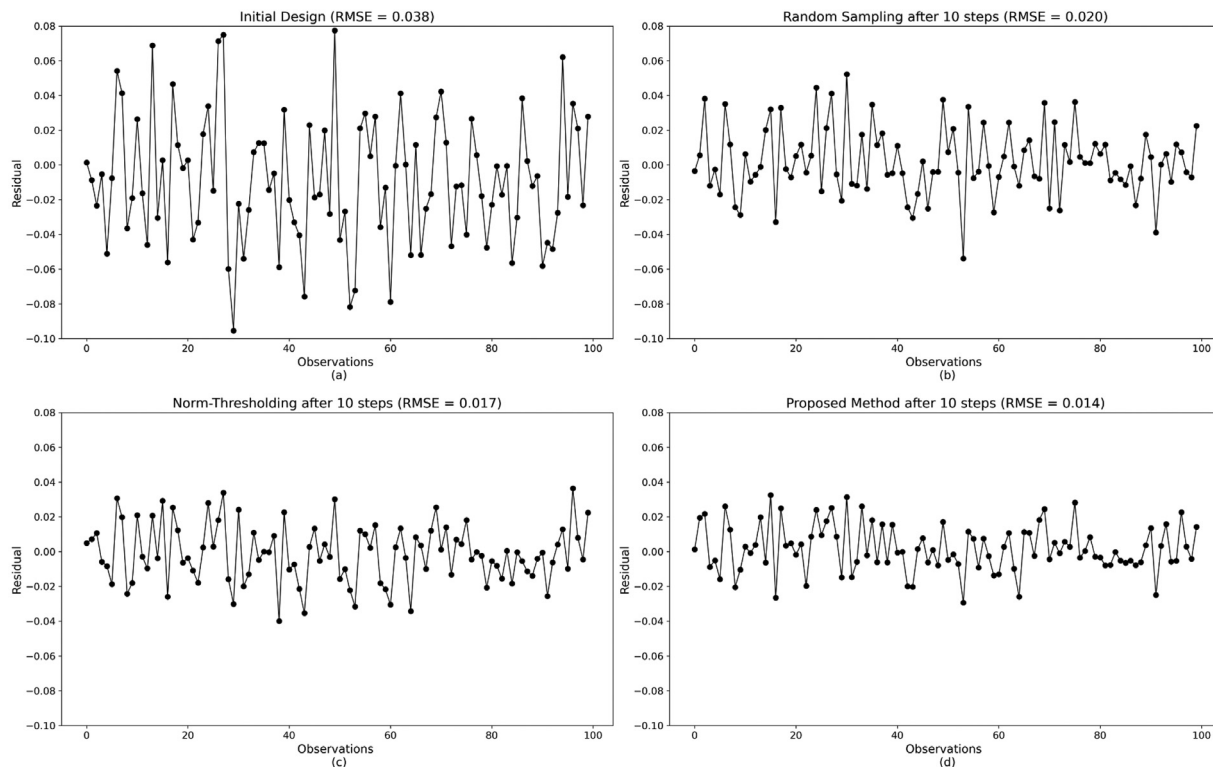
Fig. 9. Residuals of the Stream 11D predictions: with the initial training set (a) and after augmenting the design with 10 additional labeled examples with the different methods (b–d) (one simulation with $\alpha = 1\%$).

when more observations are included in the design. However, the predictions obtained with the proposed strategy are significantly better than the ones obtained with random sampling and norm-thresholding. Indeed, it should be noted how the RMSE obtained with the fifth model using CDO is 55 percent lower than the RMSE obtained with random sampling, and 23 percent lower than the RMSE obtained with the alternative active learning scheme. Finally, the improvement of CDO from the initial RMSE is higher than 65 percent.

It should be noted how a simple linear regression model fitted on a small training set can achieve compelling prediction results when the labeled examples are appropriately selected. This is true even when testing our approach on data from the TEP, which is characterized by highly nonlinear relationships.

Fig. 9 shows the predictions obtained for stream D of the product. In this case, to offer an additional view, we compared the models obtained after 10 learning steps. It can be seen how the behavior of the different schemes follows the same trend observed in Fig. 8. Indeed, after 10 iterations, the RMSE obtained with CDO is 18 percent lower than the one obtained by norm-thresholding and 30 percent lower than the one obtained with random sampling. From the initial design, the RMSE is reduced by more than 60 percent with CDO.

5. Conclusion

In many industrial processes and real-life applications, data is often abundant only in an unlabeled form. Moreover, the prohibitive cost required by quality inspections and the time required by manual annotation makes it unfeasible to label each data point with its quality characteristic. In these cases, active learning can significantly improve the predictive performance of regression models by smartly selecting the instances to include

in the training set. In situations where many observations are sequentially processed, it is necessary to provide a real-time sampling strategy for selecting the most informative instances. In this paper, we propose an optimal strategy for performing stream-based active learning with linear regression models. Two case studies, one using numerical simulations and the other one using the TEP, show that the proposed approach offers improved predictive performance and reduces the prediction error faster.

CRediT authorship contribution statement

Davide Cacciarelli: Conceived and designed the analysis, Contributed data or analysis tools (implementation of the computer code), Performed the analysis, Wrote the paper. **Murat Kulahci:** Supervision, Performed the analysis, Wrote the paper. **John Sølve Tyssedal:** Conceived and designed the analysis, Provided guidance, Supervision, Performed the analysis, Wrote the paper.

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.

Appendix

The plots in Figs. 10–13 report the learning curves showing the RMSE values, without using random sampling as baseline. The plots begin from the third learning step to better show the differences between the curves. As all the models start from the same random design, the RMSE obtained in the first learning step is the same for the three methods as it is shown in Figs. 3, 4 and 6, 7.
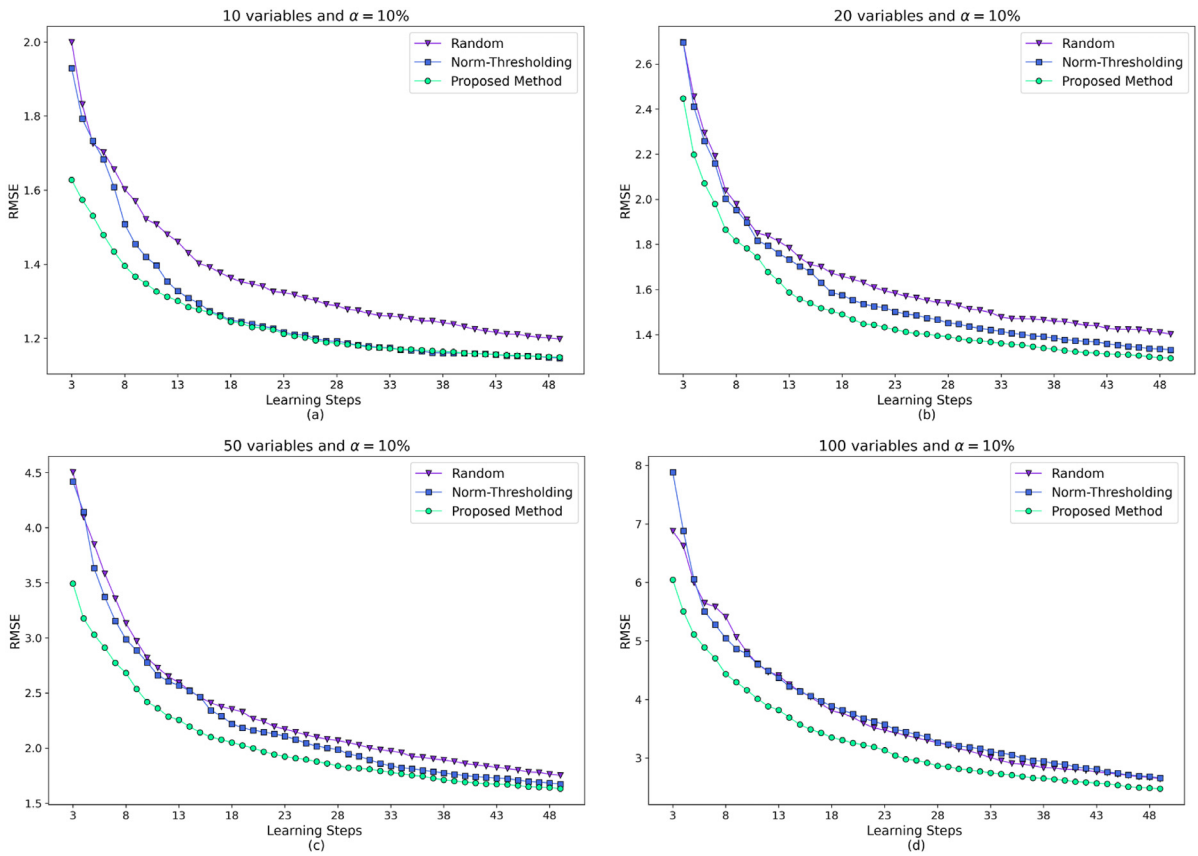
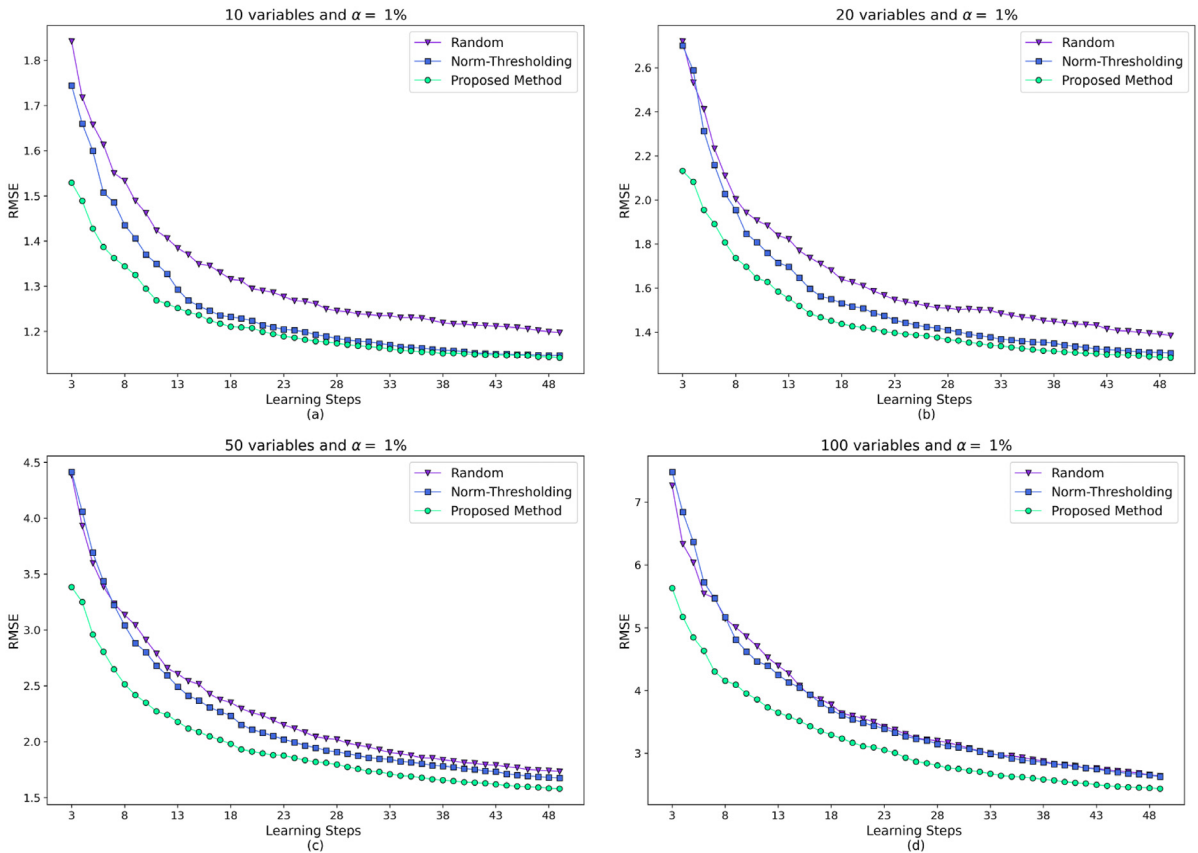**Fig. 10.** Learning curves of different methods on numerical simulations with $\alpha = 10\%$ (50 simulations).

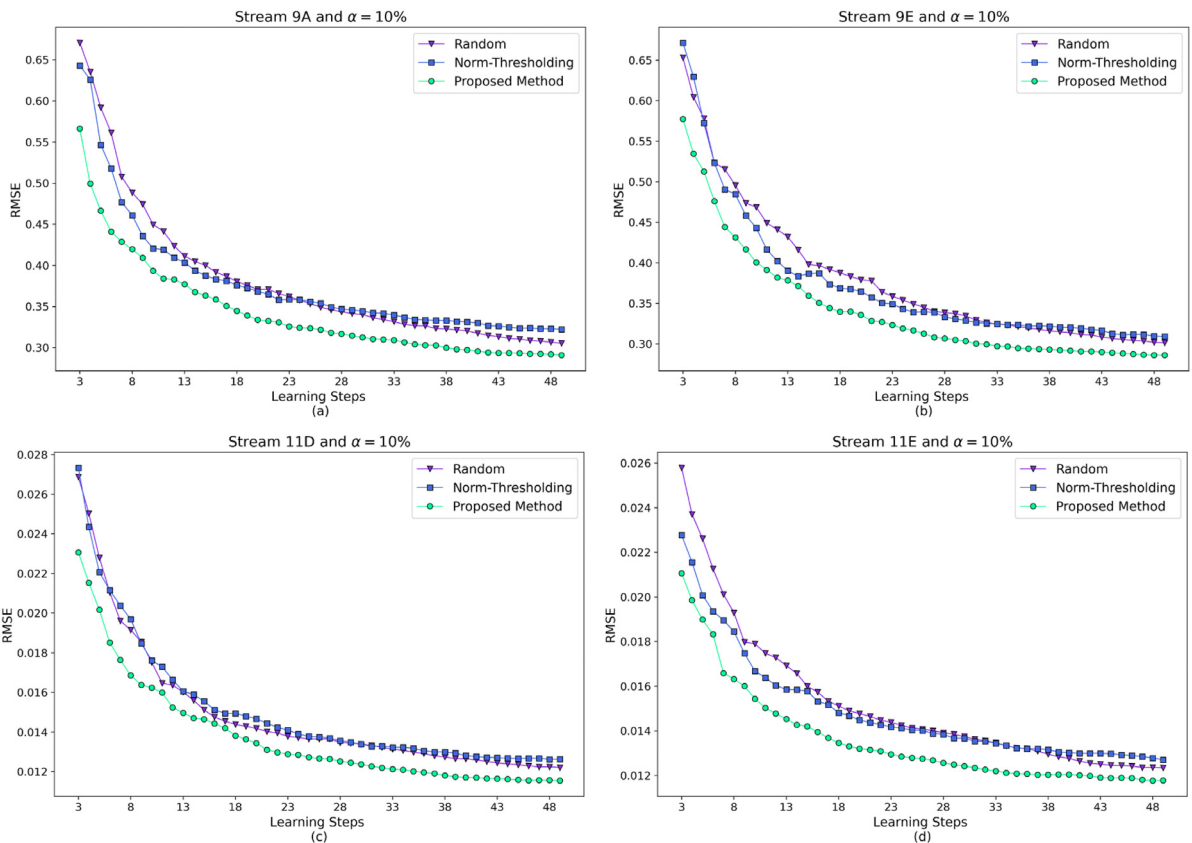**Fig. 11.** Learning curves of different methods on numerical simulations with $\alpha = 1\%$ (50 simulations).

Fig. 12. Learning curves of different methods on TEP data with $\alpha = 10\%$ (50 simulations).
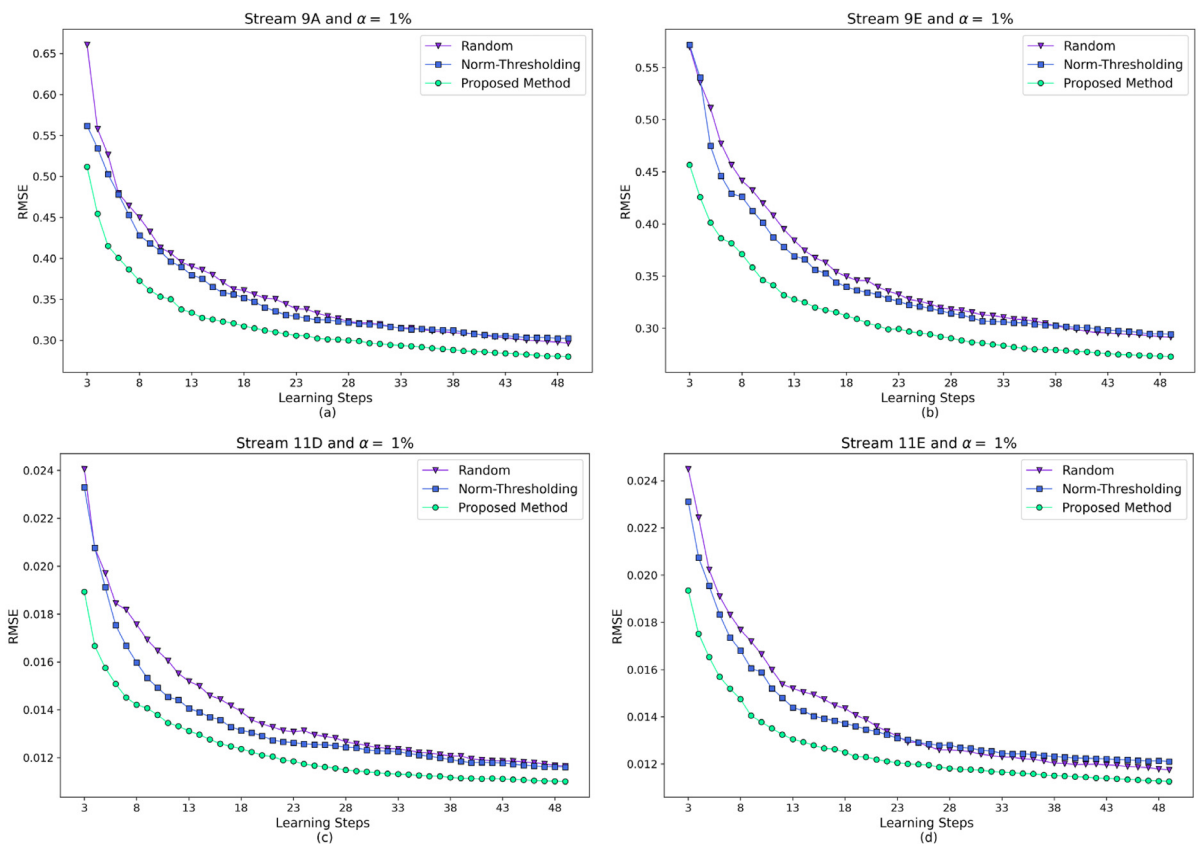

Fig. 13. Learning curves of different methods on TEP data with $\alpha = 1\%$ (50 simulations).

References

[1] P. Kumar, A. Gupta, Active learning query strategies for classification, regression, and clustering: A survey, J. Comput. Sci. Technol. 35 (2020) 913–945, http://dx.doi.org/10.1007/s11390-020-9487-4.

[2] B. Settles, Computer sciences department active learning literature survey, 2009.

[3] D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, 1994.

[4] D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning, Mach. Learn. 15 (1994) http://dx.doi.org/10.1007/BF00993277.

[5] P.R. Freeman, The secretary problem and its extensions: A review, 1983.

[6] E. Lughofer, On-line active learning: A new paradigm to improve practical useability of data stream modeling methods, Inform. Sci. 415–416 (2017) 356–376, http://dx.doi.org/10.1016/j.ins.2017.06.038.

[7] L.L.T. Chan, Q.Y. Wu, J. Chen, Dynamic soft sensors with active forward-update learning for selection of useful data from historical big database, Chemometr. Intell. Lab. Syst. 175 (2018) 87–103, http://dx.doi.org/10.1016/j.chemolab.2018.01.015.

[8] X. Shi, W. Xiong, Approximate linear dependence criteria with active learning for smart soft sensor design, Chemometr. Intell. Lab. Syst. 180 (2018) 88–95, http://dx.doi.org/10.1016/j.chemolab.2018.07.009.

[9] Z. Ge, Active learning strategy for smart soft sensor development under a small number of labeled data samples, J. Process Control. 24 (2014) 1454–1461, http://dx.doi.org/10.1016/j.jprocont.2014.06.015.

[10] Q. Tang, D. Li, Y. Xi, A new active learning strategy for soft sensor modeling based on feature reconstruction and uncertainty evaluation, Chemometr. Intell. Lab. Syst. 172 (2018) 43–51, http://dx.doi.org/10.1016/j.chemolab.2017.11.001.

[11] D. Macciò, Local linear regression for efficient data-driven control, Knowl.-Based Syst. 98 (2016) 55–67, http://dx.doi.org/10.1016/j.knosys.2015.12.012.

[12] D. Alvarez Melis, T. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2018, https://proceedings.neurips.cc/paper/2018/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf.

[13] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2004) http://dx.doi.org/10.1214/009053604000000067.

[14] C.-X. Zhang, J.-S. Zhang, Q.-Y. Yin, Early stopping aggregation in selective variable selection ensembles for high-dimensional linear regression models, Knowl.-Based Syst. 153 (2018) 1–11, http://dx.doi.org/10.1016/j.knosys.2018.04.016.

[15] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (2015) 5659–5670, http://dx.doi.org/10.1109/TIP.2015.2487860.

[16] C. Hong, J. Yu, D. Tao, M. Wang, Image-based 3D human pose recovery by multi-view locality sensitive sparse retrieval, IEEE Trans. Ind. Electron. (2014) 1, http://dx.doi.org/10.1109/TIE.2014.2378735.

[17] J. Yu, M. Tan, H. Zhang, Y. Rui, D. Tao, Hierarchical deep click feature prediction for fine-grained image recognition, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 563–578, http://dx.doi.org/10.1109/TPAMI.2019.2932058.

[18] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (2015) 767–779, http://dx.doi.org/10.1109/TCYB.2014.2336697.

[19] C. Hong, J. Yu, J. Zhang, X. Jin, K.-H. Lee, Multimodal face-pose estimation with multitask manifold deep learning, IEEE Trans. Ind. Inform. 15 (2019) 3952–3961, http://dx.doi.org/10.1109/TII.2018.2884211.

[20] F.D. Frumosu, M. Kulahci, Big data analytics using semi-supervised learning methods, Qual. Reliab. Eng. Int. 34 (2018) 1413–1423, http://dx.doi.org/10.1002/qre.2338.

[21] L. Fortuna, S. Graziani, A. Rizzo, M.G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes, Springer, 2007.

[22] R. Burbidge, J.J. Rowland, R.D. King, Active Learning for Regression based on Query by Committee.

[23] Z. Ge, Active learning strategy for smart soft sensor development under a small number of labeled data samples, J. Process Control. 24 (2014) 1454–1461, http://dx.doi.org/10.1016/j.jprocont.2014.06.015.

[24] Z. Ge, Active probabilistic sample selection for intelligent soft sensing of industrial processes, Chemometr. Intell. Lab. Syst. 151 (2016) 181–189, http://dx.doi.org/10.1016/j.chemolab.2016.01.003.

[25] O. Reyes, A.H. Altalhi, S. Ventura, Statistical comparisons of active learning strategies over multiple datasets, Knowl.-Based Syst. 145 (2018) 274–288, http://dx.doi.org/10.1016/j.knosys.2018.01.033.

[26] W. Cai, Y. Zhang, J. Zhou, Maximizing expected model change for active learning in regression, in: Proceedings - IEEE International Conference on Data Mining, ICDM, 2013, pp. 51–60, http://dx.doi.org/10.1109/ICDM.2013.104.

[27] S. Karlin, J. William, Studden, optimal experimental designs, Ann. Math. Stat. 37 (1966) 783–815.

[28] R.H. Myers, D. Montgomery, C.M. Anderson-Cook, Response surface methodology: process and product optimization using designed experiments, 2016.

[29] R.C. st. John, N.R. Draper, D-optimality for regression designs: A review, Technometrics 17 (1975) 15–23, http://dx.doi.org/10.1080/00401706.1975.10489266.

[30] D.C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons Inc., Hoboken, NJ, USA, 2012, http://dx.doi.org/10.1002/9781118147634.

[31] X. Fontaine, P. Perrault, M. Valko, V. Perchet, Online a-Optimal Design and Active Linear Regression, 2021.

[32] C. Riquelme, R. Johari, B. Zhang, Online active linear regression via thresholding, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, www.aaai.org.

[33] E. Lughofer, M. Pratama, Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models, IEEE Trans. Fuzzy Syst. 26 (2018) 292–309, http://dx.doi.org/10.1109/TFUZZ.2017.2654504.

[34] E. Lughofer, Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, http://dx.doi.org/10.1007/978-3-642-18087-3.

[35] D.C. Hoaglin, R.E. Welsch, The Hat matrix in regression and ANOVA, Am. Stat. 32 (1978) 17, http://dx.doi.org/10.2307/2683469.

[36] G. He, Y. Li, W. Zhao, An uncertainty and density based active semi-supervised learning scheme for positive unlabeled multivariate time series classification, Knowl.-Based Syst. 124 (2017) 80–92, http://dx.doi.org/10.1016/j.knosys.2017.03.004.

[37] M.C. Fernandes, T.F. Covões, A.L.V. Pereira, Improving evolutionary constrained clustering using active learning, Knowl.-Based Syst. 209 (2020) 106452, http://dx.doi.org/10.1016/j.knosys.2020.106452.

[38] Y. Leng, X. Xu, G. Qi, Combining active learning and semi-supervised learning to construct SVM classifier, Knowl.-Based Syst. 44 (2013) 121–131, http://dx.doi.org/10.1016/j.knosys.2013.01.032.

[39] E.B. Andersen, I.A. Udugama, K. v. Gernaey, A.R. Khan, C. Bayer, M. Kulahci, An easy to use GUI for simulating big data using Tennessee Eastman process, Qual. Reliab. Eng. Int. 38 (2022) 264–282, http://dx.doi.org/10.1002/qre.2975.

[40] N.L. Ricker, Optimal steady-state operation of the Tennessee Eastman challenge process, Comput. Chem. Eng. 19 (1995) http://dx.doi.org/10.1016/0098-1354(94)00043-N.

[41] N. Lawrence Ricker, Decentralized control of the Tennessee Eastman challenge process, J. Process Control. 6 (1996) http://dx.doi.org/10.1016/0959-1524(96)00031-5.

[42] T.J. McAvoy, N. Ye, Base control for the Tennessee Eastman problem, Comput. Chem. Eng. 18 (1994) http://dx.doi.org/10.1016/0098-1354(94)88019-0.

[43] F. Capaci, E. Vanhatalo, M. Kulahci, B. Bergquist, The revised Tennessee Eastman process simulator as testbed for SPC and DoE methods, Qual. Eng. 31 (2019) http://dx.doi.org/10.1080/08982112.2018.1461905.

[44] P.R. Lyman, C. Georgakis, Plant-wide control of the Tennessee Eastman problem, Comput. Chem. Eng. 19 (1995) http://dx.doi.org/10.1016/0098-1354(94)00057-U.

[45] L. Bao, X. Yuan, Z. Ge, Co-training partial least squares model for semi-supervised soft sensor development, Chemometr. Intell. Lab. Syst. 147 (2015) 75–85, http://dx.doi.org/10.1016/j.chemolab.2015.08.002.

[46] X. Jia, W. Tian, C. Li, X. Yang, Z. Luo, H. Wang, A dynamic active safe semi-supervised learning framework for fault identification in labeled expensive chemical processes, Processes 8 (2020) http://dx.doi.org/10.3390/pr8010105.

[47] J. Zhu, Z. Ge, Z. Song, Robust semi-supervised mixture probabilistic principal component regression model development and application to soft sensors, J. Process Control. 32 (2015) 25–37, http://dx.doi.org/10.1016/j.jprocont.2015.04.015.

[48] R. Grbić, D. Slišković, P. Kadlec, Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models, Comput. Chem. Eng. 58 (2013) 84–97, http://dx.doi.org/10.1016/j.compchemeng.2013.06.014.

[49] L. Yin, H. Wang, W. Fan, Active learning based support vector data description method for robust novelty detection, Knowl.-Based Syst. 153 (2018) 40–52, http://dx.doi.org/10.1016/j.knosys.2018.04.020.

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

[51] C. Reinartz, M. Kulahci, O. Ravn, An extended Tennessee eastman simulation dataset for fault-detection and decision support systems, Comput. Chem. Eng. 149 (2021) 107281, http://dx.doi.org/10.1016/j.compchemeng.2021.107281.