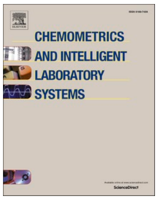
Chemometrics and Intelligent Laboratory Systems 246 (2024) 105082

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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



Variable time delay estimation in continuous industrial processes

Marco Cattaldo [a,b,*], Alberto Ferrer [b], Ingrid Måge [a]

[a] Nofima - Norwegian Institute of Food, Fisheries and Aquaculture Research P.O. Box 210, N-1431, Ås, Norway
[b] Multivariate Statistical Engineering Group, Department of Applied Statistics, Operations Research and Quality, Universitat Politècnica de València, 46022, València, Spain

ARTICLE INFO

Keywords:
Variable synchronisation
Data preprocessing
Soft-sensor
Industrial data analytics
Statistical modelling

ABSTRACT

Digital sensors and machine learning enable efficiency improvements in production processes, through process monitoring, anomaly detection, soft sensing, and process control. However, the development of such solutions requires several data preprocessing steps. In continuous processes, a crucial part of the data preparation is adjusting for time delays between different sensors. This is necessary to ensure that each sensor measurement relate to the same volume of materials going through various processing steps.

This study provides an overview of data-driven methods for estimating time lags between sensors in continuous processes. The methods are assessed in a large simulation study, on data sets with different sample sizes, model complexities and autocorrelation functions. Our results shows that most methods work well if the relationships are close to linear, but more flexible metrics like distance correlation and maximum information coefficient are needed in more complex systems. Finally, we present a real industrial example to illustrate some real-world aspects of the variable time delay estimation process.

1. Introduction

With the increasing affordability of physical and digital sensors, considerable process data has become available from most production processes. Along with the amount of data, increased awareness of the importance of data in processes fostered an increased effort in research and application of multivariate statistical and machine learning methods in various industrial settings [1,2]. Researchers and industrial practitioners started to make use of the large amounts of data originating from industrial processes at most levels [3], helping, among other things, e.g., decision-making [4], production planning and control [5], fault detection and prediction [6], predictive maintenance [7], energy efficiency [8], and quality control [9], to cite a few. In some applications, there is a need to obtain a frequent and reliable prediction as a surrogate of a quantity that is impractical, impossible, or time-consuming to measure. To obtain this prediction usually, models linking process variables and process states to the quantities of interest are used to predict it, e.g., powder composition in continuous tablet manufacturing [10], product quality in a batch polymerisation process [11], and concentrations in the bottom part of a distillation column [12]. These predictive models are often called inferential sensors, virtual online analysers, observer-based sensors, or, more commonly, soft sensors, from a union of the words “software” and “sensor” [13,14].

These soft sensors can also be categorised by the amount of physical and process knowledge the researchers and practitioners decide to include in the underlying model. When the soft sensor models are based on fundamental laws governing the relationships among the system’s physiochemical, biochemical, and mechanical properties, these are called first principles, mechanistic, or white-box models. When they are based on relationships derived from the available data, they are called data-driven or black-box models. When they are built employing both, with one class of models benefitting from the results of the other, they are called grey-box or hybrid models [15–18]. Notwithstanding the choice of model structure, soft sensors require data of sufficient quality to provide accurate and reliable predictions. Industrial data often suffer from low quality in the form of missing data, outliers or varying noise levels, and need some specific steps to increase their overall quality [19]. These steps are referred to collectively as pre-processing. They are among the most time-consuming steps in a model-building pipeline, taking up as much as 75% of a practitioner’s time during model-building [20].

A generally overlooked problem in data pre-processing is variable time delay [21]. A delay often occurs between the target and input variables in real industrial processes. This delay may depend on various

[*] Corresponding author. Nofima - Norwegian Institute of Food, Fisheries and Aquaculture Research, P.O. Box 210, N-1431, Ås, Norway.
E-mail address: marco.cattaldo@nofima.no (M. Cattaldo).

https://doi.org/10.1016/j.chemolab.2024.105082
Received 13 November 2023; Received in revised form 25 January 2024; Accepted 30 January 2024
Available online 9 February 2024

sources but can generally be classified as either measurement/signal delay or process delay [22]. Measurement delay is due to sensors taking time to measure some quantity, while process delay is a characteristic of the process and comes from, e.g., the time it takes from a given set of materials to go from a unit operation to the next, the residence time in a reactor or the blend time in a mixer. Variable time delay substantially impacts process modelling, especially in continuous industrial processes where there is a need to join data from different sensors that are distant in space and need to be adjusted for if an accurate model is desired [21]. Failing to address this delay might lead to inaccurate models especially in the case of process that are highly variable over time or when the process is not running at steady state.

Process technicians often have some knowledge about approximate time delays between measurement points but are not always accurate enough to obtain good modelling results, as estimating flows from first principles can be especially challenging when conditions deviate from ideal, for example when buffer tanks are present where laminar flow cannot be expected or when parallel processing units create a mixture of flows. The time delay may also vary depending on changes in the material flow or process settings. It is, therefore, often necessary to estimate the variable time delay from data. Over the years, several papers have been published on variable time delay estimation using different techniques and approaches. However, to the best of our knowledge, an extensive overview and comparative study has yet to be carried out on these methods. In this paper, we compare different approaches for variable time delay estimation on industrial and synthetic datasets. We evaluate the methods' ability to estimate the correct time delay in different scenarios, their computation time, and general ease of use.

The rest of the paper is organised as follows. Section §2 will clarify the variable time delay problem with simple mathematical notation. Section §3 will overview the estimation methods used in the comparison. In section §4, the datasets used for the comparison are introduced. In section §5, the results are assessed and discussed, and based on the results, we give recommendations and guidelines for estimating variable time delay in continuous industrial processes.

2. Variable time delay

This section aims to give a brief mathematical formulation of the issue. Yao and Ge [23] have an excellent explanation of the problem, and we refer to that paper for further doubts on the topic.

The aim of a soft sensor is to predict a quality attribute from a set of explanatory variables. The prediction model is usually made by training a machine learning algorithm on a set of data. Let X be a matrix of N observations (rows) with K columns $(\mathbf{x}_1, \dots, \mathbf{x}_K)$ containing the values of K explanatory variables $(\mathrm{X}_1, \dots, \mathrm{X}_K)$ sampled at regularly spaced time points in a continuous industrial process where variable time delay is present, and y be an $N \times 1$ vector of a critical quality attribute Y measured from the same process.

Let f be some function linking X to y,

$$\mathbf{y} = f(\mathbf{X}) + \mathbf{e}, \tag{1}$$

where **e** is an error vector.

Introducing t as the time index of the series comprising X and y, we have that the above equation might not generally be valid for the naive case of comparing data at the same time index

$$y_t \neq \mathrm{f}\left(\mathbf{x}_t^T\right) + e_t, \tag{2}$$

where $\mathrm{x}_t^{\mathrm{T}}$ is a row vector containing the values of the K variables at time t.

To correctly assess the relationship between the explanatory variables and the quality attribute, it is necessary to account for the time delay between variables in X and y.

Let L^d be a shift operator [24] of order d, such as

$$L^d(x_t) = x_{t-d}, \tag{3}$$

The variable time delay estimation problem then becomes to identify the unknown order of the shift operator vector that satisfies equation (1) whilst minimising the norm of the error vector e.

What is required, in other words, is the vector d that satisfies

$$\begin{cases} L^{\mathbf{d}}\left(\mathbf{x}_t^T\right) = \left[x_{1,t-d_1}, x_{2,t-d_2}, x_{3,t-d_3}, \dots, x_{K,t-d_K}\right] \\ \boldsymbol{y} = f\left(L^{\mathbf{d}}(\mathbf{X})\right) + \mathbf{e} \end{cases} \quad \text{s.t. } \underset{\mathbf{d}}{argmin}\ \|\mathrm{e}\| \tag{4}$$

where d in equation (4) is a K × 1 vector, and each value of the vector is $\mathrm{d}_k \in [\Delta_l, \Delta_u]$, where Δ_l and Δ_u are, respectively, the lower and upper limits of the time delay, two parameters usually known by process technicians. The chosen limits may also incorporate prior knowledge about the order of sensors. Fig. 1 has a simple representation of this problem on a system where three process variables and a target quantity are measured with time delay.

It is important to note that while the chosen notation for Equation (4) and Fig. 1 might suggest that only one lag per variable can exist in the relationship with y, this is not the case; dynamics are an essential feature of most processes. In this case, it might be that X1 and X2 (in Fig. 1) come from the same sensor at t-4 and t-3 or that the subscript k in equation (4) refers to variables and their lags. In these situations, a single sensor can be present more than once at different lags. In this case, the time delay we are interested in detecting would be the one that occurs first.

On a more practical side, the variable time delay estimation procedure can be divided into four steps.

1) Set a fixed zero point on the time axis
2) Identify the time frame to be used in the analysis
3) Perform the delay analysis
4) Assess the results

Consider the alignment of two sensors. The first step requires the choice of which sensor to set as the reference time for the variable time estimation. When the end goal of the analysis is the creation of a soft sensor, the most probable target would be the dependent target variable, while for different analysis goals expert knowledge should be used to identify the appropriate zero point. The second step requires data analysis paired with domain knowledge to identify the correct time resolution of the variations in the quantities measured by these sensors and to consider the possible differences in measurement rates and granularity, in addition to the characteristic time of the process and the control system. The sampling frequency influences the dynamic order of the data; higher frequency sampling will result in higher order dynamic present just by virtue of the sampling happening faster than the variation in the measured quantity and their control system. While the almost universal assumption for data analysis is that all available records have the same resolution and time step, this is frequently not the case [25,26]. While process data is usually collected instantaneously, but possibly with different sampling rates, quality attributes often require manual sampling and therefore may have low and possibly irregular time resolution. Additionally, measures can sometimes be collated over different periods, e.g., shifts, hours or production runs. This difference in granularity needs to be considered, as the correct time resolution from a process point of view might be incompatible with the resolution or frequency of the measurements. Consequently, some sort of aggregation or interpolation is usually needed before the delay analysis can take place [27,28].

The third and fourth steps require performing and assessing the time delay estimation, which is the main topic of this work. Various methods are available for the analysis; a limited survey divided into different general approaches is reported in section §3. Regarding the assessment, process knowledge should be used to double-check the results. A general

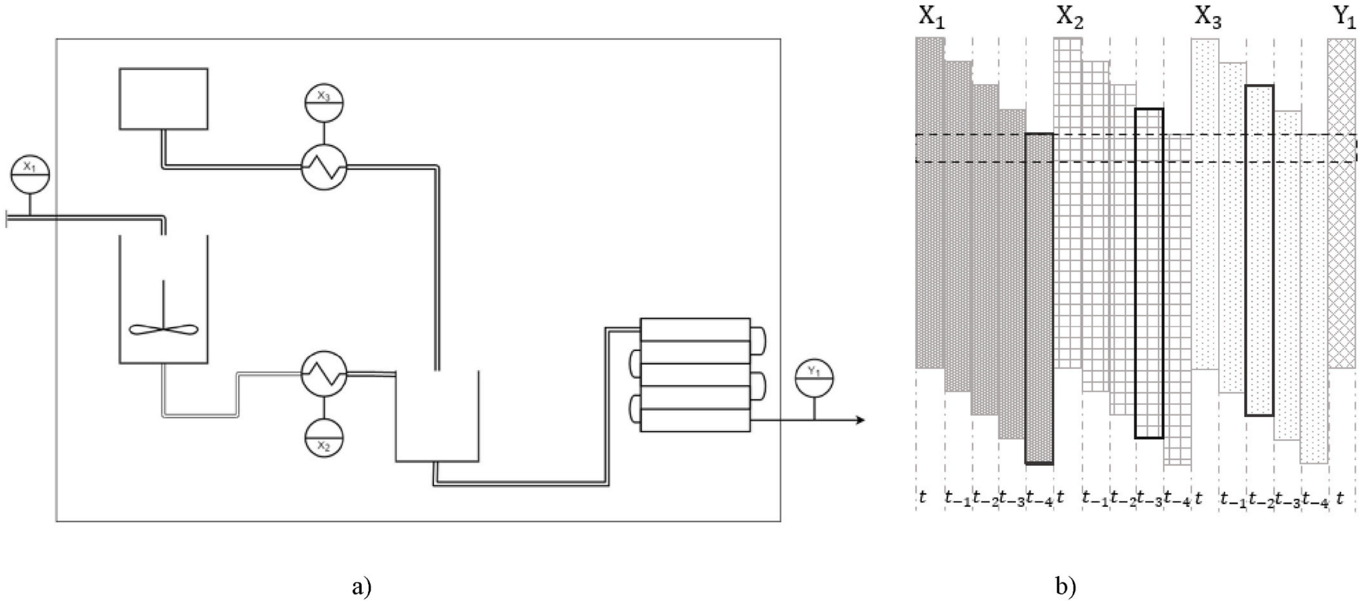


Fig. 1. Illustration of the Variable Time Delay issue: X_1, X_2 and X_3 are process variables, and Y is a quality attribute. a) shows a flowchart of the process with placements of the sensors, b) data vectors where each of the process variables are time-shifted with d = 0–4. The vectors that are marked by solid-line black rectangles are the time step that is responsible for the variation in the quality attribute highlighted by the dashed black rectangle; in this case, the **d** vector in equation (4) would be [2–4] as the variables responsible for the quality Y are $L^4(X_1)$, $L^3(X_2)$, and $L^2(X_3)$.

goal of variable time delay estimation might be the time alignment of key process parameters and raw material characteristics to the end-product quality attributes. It is worth noticing that in industrial practice, not all the measured quantities are correlated to the end-product quality, and some processing steps may break the spatiotemporal correlation structure of some or all the variables. A typical example of a step that breaks the correlation structure in *one* variable would be a heat exchanger breaking the temporal correlation in the temperature. An example of a step that breaks the correlation structure in *most* variables would be a non-FIFO (First In-First Out) step, for example, a mixing vessel, where, e.g., the temperatures and concentrations of a flow get averaged over a period. In these situations it might be useful to break down the process in smaller steps or to increase the time resolution of the analysis.

3. Variable time delay estimation methods

This section provides a survey of existing variable time delay estimation techniques and identifies the algorithms considered in the evaluation. The variable time delay estimation problem is a transversal issue that touches many different disciplines, among others, biomedical engineering [29], control systems theory [30], time series analysis [31], and chemical engineering [32], each with its preferred algorithms. While there is a partial overlap in techniques, not all can be applied to all data types. Here, we focus on those that apply to process data.

Table 1 lists all the methods evaluated in this paper, and each method is briefly described in the following subsections. All methods can be applied statically or dynamically. The dynamic approach tackles dynamic variable time delay, i.e, when the **d** vector of equation (4) is not constant thoughout the process, and generally involves a moving window [32–34].

3.1. Methods based on measures of dependence

All these methods work by calculating some measure of dependence between the response variable Y and lagged versions of the X-variables.

Table 1
Methods employed in the analysis.

Method	Abbreviation	Family	Multivariate	Main assumptions	Reference
Pearson’s correlation	r	Measure of dependence	No	Gaussian distribution and linear relationship	[35]
Kendall’s correlation	τ		No	Monotonous relationship	[35]
Spearman’s correlation	ρ		No	Monotonous relationship	[35]
Maximum Information Coefficient	MIC		No	Any kind of relationship	[36]
Mutual information	MI		Yes	Any kind of relationship	[37]
N-Norm	N-Norm		No	Any kind of relationship	[38]
Distance correlation	dCorr		Yes	Any kind of relationship	[39]
PLS regression coefficients	PLS-Beta	Variable importance in regression	Yes	Linear and some kind of nonlinear relationship	[40]
PLS regression + selectivity ratio	PLS-SR		Yes	Linear and some kind of nonlinear relationship	[41]
Kernel PLS + variable permutation	KPLS		Yes	Any kind of relationship	[42]
Random forest + Variable Permutation	RF		Yes	Any kind of relationship	[43]
Variable time reconstruction using Linear Least Squares	VTR-LS	Optimisation	Yes	Linear and some kind of nonlinear relationship	[23]
Variable Time Reconstruction using Gaussian mixtures	VTR-GMM		Yes	Gaussian mixture distribution	[23]
Genetic algorithm with mutual information	GA-MI		Yes	Any kind of relationship	[44]

The optimal lag for each X is then defined as the one having maximum dependence on the response.

3.1.1. *Classical correlation metrics*

Pearson's (r), Kendall's (τ), and Spearman's (ρ) correlations are classical methods that see widespread use in data analysis. Pearson product-moment correlation measures the strength of the linear relationship between two variables and is one of the most used statistical estimators. Kendall's and Spearman's are two other very well-known correlation indices; they measure the strength of monotone relationships between two variables using ranks (i.e., relative position label of the observations within the variable: 1st, 2nd, 3rd, …, n-th) between the two variables, resulting in increased performance and less influence from outliers, with Kendall's being the most outlier resistant of the three [45].

Some examples of these methods applied to variable time delay estimation are the pre-processing step of modelling a de-aromatisation process [37] and calculating the melt flow index in a polymerisation process [46].

3.1.2. *Mutual information*

Mutual information (MI) between two random variables, X and Y, $\mathrm{MI}(\mathrm{X},\mathrm{Y})$, measures the information that X and Y share; it is a measure capable of describing any relationship that measures the reduction in uncertainty about one variable that can be obtained by observing the other. Mutual Information is measured mathematically as the difference between the entropy of the marginal distributions of the two variables and the entropy of the joint distribution of the two variables:

$$\mathrm{MI}(\mathrm{X},\mathrm{Y}) = \mathrm{H}(\mathrm{Y}) + \mathrm{H}(\mathrm{X}) - \mathrm{H}(\mathrm{X},\mathrm{Y}) \tag{5}$$

where H(X) and H(Y) are the marginal entropies of X and Y, respectively, and H(X, Y) is their joint entropy. Marginal entropies are calculated for discrete variables as follows:

$$H(\boldsymbol{X}) = H(\mathrm{X}_1,\mathrm{X}_2,\ldots,\mathrm{X}_\mathrm{K}) = \sum_{x_1}\sum_{x_2}\ldots\sum_{x_\mathrm{K}} -P_\mathrm{X}(x_1,x_2,\ldots,x_\mathrm{K})\log(P_\mathrm{X}(x_1,x_2,\ldots,x_\mathrm{K})) \tag{6}$$

where $P_X(x)$ is the marginal probability mass function of X; and for continuous variables as:

$$H(\boldsymbol{X}) = H(\mathrm{X}_1,\mathrm{X}_2,\ldots,\mathrm{X}_\mathrm{K}) = \int_{x_1}\int_{x_2}\ldots\int_{x_\mathrm{K}} -P_\mathrm{X}(x_1,x_2,\ldots,x_K)\log(P_\mathrm{X}(x_1,x_2,\ldots,x_K))dx_1dx_2\ldots dx_K \tag{7}$$

where $P_X(x)$ is the marginal probability density function of X. The joint entropy $\mathrm{H}(\mathrm{X},\mathrm{Y})$ is calculated similarly, but the integral is on x and y, and the joint probability $P_{XY}(xy)$ substitutes for $P_X(x)$.

Mutual information is not straightforward to estimate from empirical data as the calculation of entropies is based on probabilities and suffers from possible biases [47,48].

These difficulties and biases are exacerbated by Mutual Information being derived specifically for discrete quantities, and the infinite integral of equation (7) being often impossible to solve. Nevertheless, a wide variety of strategies to estimate Mutual Information in real-valued data that account for these biases [48], albeit not universally, have been devised. The most widely used estimation strategies are based, among others, on k-nearest neighbour (k-NN) [49], kernel density estimation [50] or space discretisation [51]. MI has been extensively used for feature selection [52]; additionally, it was explicitly used for variable time delay estimation [53,54].

In this work, the KSG estimator (a k–NN–based estimator) from Kraskov et al. [49] has been used, virtually unchanged from the code provided in the MILCA package [55]. Although Mutual information is traditionally multivariate, we have employed it in a bivariate way. It is worth noting that MI is not a measure of dependence as per the definition of, e.g., Rényi [56] (i.e., it is not bounded between 0 and 1, for practical use), but when transformed with Linfoot's [57] formula, it becomes one. The transformed version has been used when comparing MI with other measures of dependence.

3.1.3. *Maximal Information Coefficient*

The Maximal Information Coefficient (MIC) is a measure of dependence between two random variables, X and Y (components of a multivariable random variable), formulated by Reshef et al. [36], useful for all kinds of relationships. The MIC is based on an algorithm that sequentially partitions the marginals of the joint distribution of X and Y. On these partitions, i.e., different ways of discretising the data, MI is calculated and then normalised by dividing by the base(2) logarithm of the grid size (i.e., the minimum of the number of times it was divided on each dimension), and the highest resulting normalised mutual information among all grids is chosen.

MIC is presented as an equitable measure of dependence, i.e., it should present the ability to rank in the same way different relationship types that present the same noise level [58,59]. However, this is still debated [60,61]. To the best of our knowledge, there are no direct examples of using MIC for variable time delay, but it has been used as a part of the N-Norm method (see section §3.1.5). In other fields, it has been used, e.g., for predicting periodic patterns in large-scale time--resolved protein expression profiles and exploring the coupling relationship between two time series in specific frequency bands [62,63]. The MATLAB version of the minepy package detailed in Albanese et al. [51] has been used in this work.

3.1.4. *Distance correlation*

Distance correlation is a multivariate measure of dependence between two paired data matrices **X** and **Y**, that can measure relationships of any kind. It was developed by Székely, Rizzo, and Bakirov and expanded upon by Székely and Rizzo [39,64,65]. Distance correlation is based on the intuition that if the distances among observations in **X** and **Y** co-vary, then **X** and **Y** are, in some way, related. Mathematically, this is done by first calculating the pairwise distances between the N observations in **X** and in **Y**, referred to as a_{ij} and b_{ij}, respectively, and then double centring,

$$\begin{matrix} A_{ij} = a_{ij} - \overline{a}_{i\bullet} - \overline{a}_{\bullet j} + \overline{a}_{\bullet\bullet}, a_{ij} = \left\|\mathbf{x}_i - \mathrm{x}_j\right\| \\ B_{ij} = b_{ij} - \overline{b}_{i\bullet} - \overline{b}_{\bullet j} + \overline{b}_{\bullet\bullet}, b_{ij} = \left\|\mathbf{y}_i - \mathbf{y}_j\right\| \end{matrix}, i,j = 1,\ldots,N \tag{8}$$

where $\overline{a}_{\bullet\bullet}$ is the overall mean, while $\overline{a}_{j\bullet}$ and $\overline{a}_{\bullet k}$ are the row and column mean, respectively, of the distance matrix between observations in **X** (and similarly for **Y**).

The distance covariance is then calculated as the average of the element-wise product of the two double-centred distance matrices,

$$dCov^2(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{N^2}\sum_j\sum_k A_{jk}B_{jk}. \tag{9}$$

The distance correlation is then calculated as in equation (10) if the denominator is above zero and is zero otherwise.

$$dCorr(\boldsymbol{X},\boldsymbol{Y}) = \frac{dCov^2(\boldsymbol{X},\boldsymbol{Y})}{\sqrt{dCov^2(\boldsymbol{X},\boldsymbol{X})\,dCov^2(\boldsymbol{Y},\boldsymbol{Y})}}. \tag{10}$$

In contrast to Pearson's correlation, which can be negative, distance correlation is bounded between 0 and 1, as MIC.

Distance correlation has been used for time series autocorrelation identification [66], identification of associations in astrophysical databases [67], in a version of Independent Component Analysis [68], and a non-parametric extension of ANOVA [69]. Although distance correlation is generally multivariate, it is used here as a bivariate measure of dependence. The code used in this work has been written by the authors and tested for consistency against the implementation of Shen et al. [70].

3.1.5. Panel-based methods

N-Norm is a class of methods that employs a Euclidean norm to aggregate several different measures of dependence to identify variable time delay; it was introduced by Graziani and Xibilia [71] and fleshed out in further publications by the same authors [38,72]. The application we replicate here uses an expert panel composed of Pearson's, Kendall's, and Spearman's correlations and Maximal information coefficient [38]. Unlike most other variable time delay estimation techniques, the application of this method is based on the lag augmentation of the response vector **y**. While all the N-Norm-type methods are worthy of consideration and the base framework described in the referred publications is flexible and can be adapted for many situations, it is specifically tailored for applications where many **X** variables have the same delay from **y**. This is not the case of the synthetic dataset of §4.1, nor for the industrial example of §4.2. When applied to a dataset where only one variable X is paired to the vector **y** the measures of [71,72] simplify to Person correlation included in §3.1. The measures of dependence among the lagged y and the input X variables are collected in a vector, and their norm is calculated. The norm is then used to rank the dependence of the inputs on the column of the augmented **y** response vector. In this work, the N-Norm was coded using base MATLAB functions for the classical correlations and minepy [73] for MIC.

3.2. Methods based on multivariate regression

Estimating time delay can be seen as a sister task to variable selection in multivariate regression. In this case, an augmented descriptor matrix is created by adding lagged copies of each X-variable. Contrary to classical variable selection, a restriction to select *one* variable for each subset of lagged variables (corresponding to the same X variable) is defined. It could, therefore, be possible to adapt, with some changes, most of the variable selection literature to this task. That said, a wealth of publications comparing and rating variable selection algorithms are available [74–76], but it is beyond the scope of this paper to include all. We have focused on some of the most popular regression methods offering well-established metrics for variable importance. The optimal lag for each X-variable is then defined as the lag with the highest variable importance within each subset of lagged variables.

3.2.1. Partial Least Squares Regression

Partial Least Squares Regression (PLSR) is a widely used multivariate regression and dimensionality reduction tool between two sets of variables. This technique is based on the projection of the original data matrices (**X** and **Y**) into a lower-dimensional space called latent space, where the regression modelling is carried out. Mathematically, the direction of the latent space is defined to maximise the covariance of the latent variables (linear combinations of the original variables) between both spaces [77,78]. While PLSR is not a selection method, many feature selection algorithms have been proposed based on it; see Refs. [79,80] for a review. Most of these methods can be adapted to variable time delay estimation. In this comparison, the Selectivity Ratio and Beta coefficients have been selected based on the information from Mehmood et al. [79]. Selectivity Ratio is a method for variable importance based on target projection. It uses the beta coefficient of the estimated model to recalculate the predictors' weight and, from that, recalculates loadings and scores. These new loadings can be used to calculate how much each predictor variable contributed to the response of the estimated model. Further information can be found in the review cited above and the paper first describing the method [41,79].

Beta coefficients use the absolute value of the regression coefficients of the estimated latent variable model to gauge the importance of each X-variable in the predictor set.

3.2.2. Kernel Partial Least Squares Regression

Kernel partial least square (KPLS) is a nonlinear extension of the PLSR algorithm based on the kernel transformation. The kernel transformation uses a so-called kernel function that projects the original **X** matrix into an increased dimensional space, called feature space, where it is possible to describe in linear ways relations that would have been nonlinear in the original space. Additionally, it provides ways of calculating inner products in the implicit feature space without explicitly transforming the original variables; this is referred to as the kernel trick and is at the heart of the usefulness of kernel methods. Many kernel types have different functions and properties; a review can be found here [81]. In the past, kernel PLS has been used for fault diagnosis in batch and continuous processes or analysing mixture design of experiments and feature selection [82–85].

In this work, the Gaussian kernel was employed, applied through the radial basis function (RBF). Optimisation of the meta-parameter associated with the RBF has been carried out in the low-size, high-complexity cases (see Table 2 in §4.1); a value of one was selected. The authors of [76] provided the code employed. Although pseudo-samples are commonly used to gauge the importance of variables when kernel transformations are applied [86,87], the variable importance has been measured as the increase in prediction error of cross-validation incurred when removing from **X** the information of one variable at the time by randomly permuting its rows, as detailed, e.g., by Fisher et al. [88]. The variables that resulted in the highest error increase were selected.

3.2.3. Random forest

Random forest is a general-purpose classification and regression algorithm based on an ensemble of simple tree models [43]. In the standard implementations, the method combines many randomised decision trees via either polling or averaging, depending on the desired application. These randomised trees operate according to a "divide and conquer" strategy; each tree is built on a subsample taken randomly with replacement, and aggregated as described above; this is generally referred to as Bagging [89]. Random forests are considered very simple to use, with limited parameters to tune a good accuracy in a wide range of applications and the ability to deal with small sample sizes and high dimensional predictor spaces. A review of the method can be found in Ref. [90].

In this work, we used the MATLAB implementation of random forests. There are several ways of assessing variable importance in random forest models. One way is to calculate the improvement in the split criterion when a regressor variable is used in a tree split, obtaining the overall variable importance as the average over all trees in the forest. Another way of constructing the variable importance measure is using variable permutation on the out-of-bag samples, i.e., the samples not used to grow each specific tree [91]. In this work, the importance has been calculated with the same algorithm used for KPLS (§3.2.2).

3.3. Methods based on optimisation frameworks

Methods in this category work by finding the vector **d** in equation (4) either by solving the optimisation problem presented therein or by finding the vector that gives the maximum multivariate dependence between **X** and **y**, i.e., the strength of f without any assumption on the functional relation.

3.3.1. Variable time reconstruction

Variable time reconstruction (VTR) is a framework introduced by Yao and Ge in 2020 [23]. It consists of two nested loops, an outer

Table 2
Factor levels used in the analysis.

Factors	Levels		
Size	*100*	*1000*	*10000*
Complexity	*Linear*	*Polynomial*	*Power ratios*
Autoregressive Order	*1*	*2*	
Autoregressive Strength	*High*	*Low*	

modelling loop and an inner optimisation loop, used to solve a version of equation (4). It starts with a model chosen to represent the f in (4); this is the outer modelling loop. The model is then substituted into equation (4), and the optimiser finds the optimal vector $\mathbf{d}$ for the model built in the outer loop; this is the inner optimisation loop. These two loops continue iteratively until the maximum number of iterations for each is reached. The framework is very flexible; in the original paper, it is suggested that it can be used with most models as the outer modelling loop to adapt to different requirements of the modelling exercise. Outside of Yao and Ge, it has been used with an encoder-decoder network based on long short-term memory cells to extract time delay and dynamic features from semi-supervised process data to predict the output quality [92].

Here, we use the framework with two different model alternatives in the outer loop, both introduced in the original paper of Yao and Ge. These are a Linear Least Square model (VTR-LS) and a Gaussian Mixture Model (VTR-GMM). The code for the implementation of the VTR framework was obtained from the authors' GitHub.

3.3.2. *Genetic algorithm with mutual information*

This method uses a genetic algorithm to build multivariate realisation of the Mutual Information (GA-MI) reported in section §3.1.2. This version is a modified version based on the one detailed by Ludwig et al. [44], where it was used for feature selection. The method uses a genetic algorithm to identify the optimal $\mathbf{d}$ vector in equation (4). Instead of assuming the functional form f linking $\mathbf{X}$ and $\mathbf{y}$, it only assumes that the relationship between the variables can be correctly rated. The algorithm works by generating tentative vectors $\mathbf{d}_{trial}$, and calculating the multivariate dependence between $L^{\mathbf{d}_{trial}}(X)$ and $\mathbf{y}$. Algorithmically, the joint conditional entropy $H(Y|X) = H(X,Y) - H(X)$ is estimated using discretisation with equal size bins, whose number is set by a meta parameter. To avoid calculating the multidimensional probability (mass or density) function necessary for the entropy calculations, the optimisation in this algorithm is based on the maximal relevance minimal redundancy approach [93], which aims to jointly maximise the average bivariate mutual information between the single variable vectors $\mathbf{x}_k$ (see equation (11)) and $\mathbf{y}$ and minimise the mutual information among all the variable vectors in $\mathbf{X}$ (see equation (12)).

$$\max \frac{1}{n}\sum_k \mathrm{MI}\left(\mathrm{L}^{\mathbf{d}_k}(\mathrm{X}_k), \mathrm{Y}\right) \tag{11}$$

$$\min \frac{1}{n^2}\sum_k\sum_l \mathrm{MI}\left(\mathrm{L}^{\mathbf{d}_k}(\mathrm{X}_k), \mathrm{L}^{\mathbf{d}_l}(\mathrm{X}_l)\right) \tag{12}$$

The authors in the original paper [44] give proof of the equivalence between the maximal relevance and minimum redundancy approach and the minimisation of the joint conditional entropy. It is worth noting that in the aforementioned paper the equivalence holds only for independent X_k which is not the case in Variable time delay estimation. The approximation still results in acceptable performance. This work employs a modified version of the code from Ludwig et al. [44].

4. Datasets

This section presents the datasets used in the comparison. At first, synthetic datasets are used to compare the performances of the methods on datasets with different characteristics and where the ground truth is known. Afterwards, a selection of the best-performing methods are also used on industrial data to illustrate some real-world aspects of the variable time delay estimation process.

4.1. *Synthetic dataset*

Data has been generated from a simulated continuous process illustrated in Fig. 2. The response variable Y is a function of two underlying phenomena, $\mathscr{V}_1$ and $\mathscr{V}_2$. In addition, there is a third underlying phenomenon $\mathscr{V}_3$ that does not influence Y. These three phenomena are indirectly measured by three sensors each, placed at different locations along the process, represented by x's in Fig. 2. The true time delays between the response and each of the relevant sensors are $\mathbf{d} = [10,15,25,30,40,45]$.

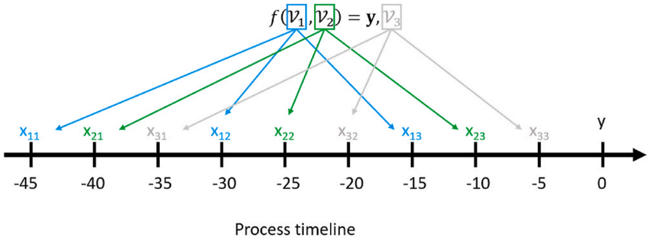


Fig. 2. Simple illustration of the simulated continuous process from which the synthetic data are generated. The x's are surrogate measurements of the real phenomena $\mathscr{V}_i$, of which the first two are used to generate the response variable y.

Data sets with different sample sizes (three levels), functional complexity (three levels), autoregressive order (two levels) and autoregressive strength (two levels) were generated according to a full factorial design; see factor levels in Table 2. The design was replicated three times, yielding a total of 108 simulated data sets.

In practical terms, a set of three vectors $\mathscr{v}_i, i \in [1,3]$ were generated from an uncorrelated normal distributed random stream $\mathscr{x}$, with zero mean and identity covariance matrix, of these only $\mathscr{v}_i, i \in [1,2]$ are correlated with y while $\mathscr{v}_3$ is uncorrelated with the response variable and is included as "noise" for the models. This stream was then passed through a one-dimensional filter to apply the selected autoregressive relationship to the data in the following form:

$$\mathscr{v}_i(t) = a_1\mathscr{v}_i(t-1) + a_2\mathscr{v}_i(t-2) + \mathscr{x}(t) \tag{13}$$

where a_i $(i = 1,2)$ measures the strength of the autoregressive behaviour and $a_2 = 0$ when autoregressive order is one.

Three "sensor" measurements are generated from each autoregressive variable by multiplication by a set of coefficients b plus a measurement error, $\epsilon \sim N(0, 0.05)$

$$\mathbf{x}_{i,j} = \mathscr{v}_i b_{i,j} + \boldsymbol{\epsilon}_{i,\mathbf{j}}, j \in [1,3] \tag{14}$$

where $x_{i,j}$ is the jth "sensor" measurement of the "true" variable $\mathscr{v}_i$.

The responses are generated with increasing order of complexity. These complexity levels are:

1) Linear

$$\mathrm{y} = \mathscr{V}_{1,2}\mathrm{l}, \mathrm{l} \sim \mathrm{N}(\mu_\mathrm{l}, \Sigma_\mathrm{l}), \tag{15}$$

where $\mu_\mathrm{l} = [-0.7071, 0.7071]^\mathrm{T}$ and $\Sigma_\mathrm{l} = \sigma\mathbf{I}$ and $\mathscr{V}_{1,2} = [\mathscr{v}_1\ \mathscr{v}_2]$.

2) Polynomials

$$\mathbf{y} = \mathscr{V}_{1,2}\mathbf{p}_{1,2} + \mathscr{v}_1^2\mathrm{p}_3 + \mathscr{v}_2^2\mathrm{p}_4 + \mathscr{v}_1\mathscr{v}_2\mathrm{p}_5, \mathbf{p} \sim \mathrm{N}[\mu_\mathrm{P}, \Sigma_\mathrm{P}] \tag{16}$$

where $\mu_\mathrm{p} = [-0.2300, 0.2300, -0.7146, -0.4732, 0.3995]^\mathrm{T}$,
$\mathrm{diag}(\Sigma_\mathrm{p}) = [0.1\sigma, 0.1\sigma, \sigma, \sigma, \sigma]$, and all off-diagonal elements are zero, $\mathbf{p}_{1,2}$ is a column vector containing the first two values of $\mathbf{p}$.

3) Power ratios

$$\mathbf{y} = \frac{(\nu_2 l_2)^2}{1 + (\nu_1 l_1)^3}, \mathbf{l} \sim \mathrm{N}(\mu_L, \Sigma_L), \tag{17}$$

where $\mu_l = [-0.7071, 0.7071]^T$ and $\Sigma_l = \sigma\mathbf{I}$, and l_i is the *i*th element of **l**.

In equations (15)–(17), σ is a dispersion parameter used to differentiate between replicates of the simulated datasets and is set to $\sigma = 0.01$. Responses are then normalised using the median and interquartile range, and a small error, $\epsilon \sim \mathrm{N}(0, 0.25)$, is added to simulate measurement error.

4.2. Kamyr dataset

The KAMYR dataset contains data from a pulp manufacturing process employing a Kamyr digester. For an in-depth description of the process and the measured variables, we refer to previous publications [94,95]. The dataset contains 301 observations of 21 predictor variables and one target variable. The target variable is the Kappa number, which characterises the pulp's quality in terms of lignin content. The lignin content determines the suitability of the pulp for different applications and, thus, is a key quality parameter in the process. The predictors are measurements from throughput and operating variables in the digester, measured hourly for eight months under closed loop. The dataset is available online at [https://openmv.net/info/kamyr-digester (accessed on 20/12/2023)]. The variable time delay was known for this dataset, and the available data is already lagged to account for it. In order to test the methods in a closer-to-reality scenario, artificial lags have been introduced in all the variables; thus, the true lags are known. The True lags for the variables and the end product quality is 11, while the upper and lower limits are 21 and 1, respectively. After adjusting for missing values, removing variables that contained more than 10% missing entries, and accounting for the introduced lags, the analysed dataset contains 19 variables and 245 observations.

4.3. Industrial dataset

A dataset taken from an industrial bioprocess is used as an example of variable time delay estimation on industrial data. In this process, the rest raw materials from poultry are upcycled into a protein product using enzymatic hydrolysis. The process is carried out continuously through a series of unit operations. Raw materials are transported from a neighbouring slaughterhouse in pipes and ground to a paste. Then, water is added to improve flowability and provide a medium for the reaction. After heating, an enzyme is added to the fluid in a mixing tank. The fluid then flows through a pipe reactor and, in the end, passes through a heat exchanger for enzyme inactivation. The resulting emulsion is then separated into a protein-rich water phase, an oil phase, and solid sediment. The water phase, the product of interest in this project, can be dried to different extents, either to a protein concentrate or to a protein powder.

Fig. 3 shows a sketch of the process along with the most relevant sensors and their rough spatial positioning; the sensors are described in Table 3.

While the aim of our analysis is to estimate time delays between each unit operation and the sampling point for the end-product quality, this cannot be achieved in "one shot" in this case as we did for the synthetic

Table 3
Process measurements and their characteristics.

Name	Description	Measurement unit	Frequency
F1	Inbound raw material flow	m^3/h	~2/s
F2	Inbound water flow	m^3/h	~2/s
F3	flow outbound from the reagent mixing tank	m^3/h	~2/s
P1	F3 pump torque	*Nm*	~2/s
P2	flow reactor pressure	*bar*	~2/s
T1	Flow reactor temperature 1	°*C*	~1/s
T2	Flow reactor temperature 2	°*C*	~1/s
T3	Flow reactor temperature 3	°*C*	~1/s
NIR	Near infrared spectra of the inbound raw material	AU	~2/s
Brix	Brix Degrees of the protein mixture	°*Brix*	Manual, ~20/d
Average molecular weight (AMW)	Average Molecular Weigh of the protein product	Dalton	Manual, ~20/d
Collagen	Collagen content in the protein product	%	Manual, ~20/d

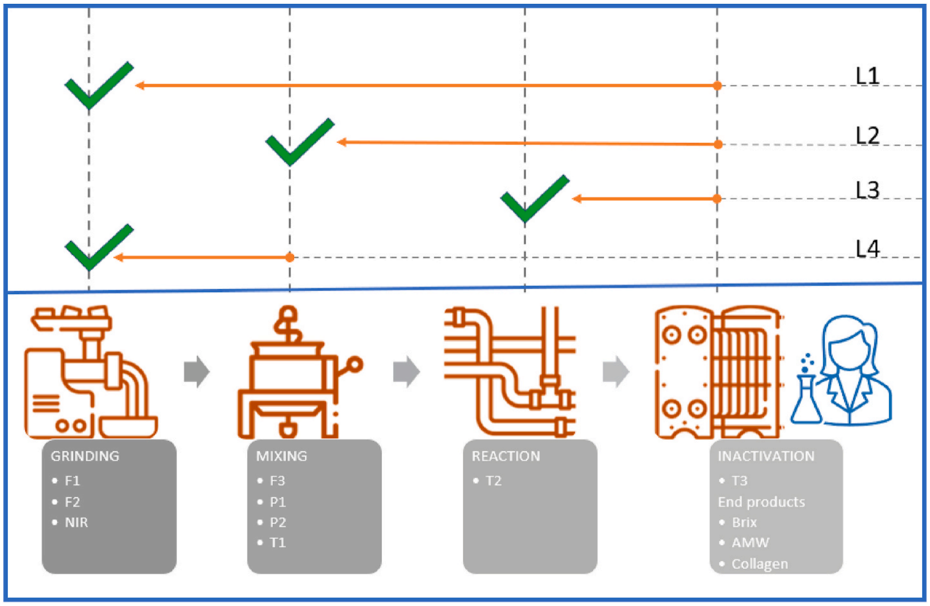


Fig. 3. Sketch of the enzymatic protein hydrolysis process and a representation of where the various sensors and sampling points are placed. The upper square shows the four alignment tasks performed in order to determine the time delays between all unit operations of this particular process.

data since the assumption that all process variables are related to the end-product quality is not met. Therefore, estimation of the four delay values (as illustrated in Fig. 3) are split into three separate estimation tasks:

1) L1: The raw materials flow and chemical composition (measured by NIR) have a strong influence on the end-product quality. For estimating the lag L1 between grinding and end-product quality, measurements in the first unit operation (grinding) are therefore used as explanatory variables, and the end qualities as responses.
2) L2 & L3: While the reaction temperature may also affect the end-product quality, the relationship is weak and does not provide precise lag estimations. The three temperature sensors are however strongly correlated as there is no exogenous heat sources between the mixing and inactivation units. The temperature sensors are therefore well suited for precise estimations of the lags L2 and L3. The temperature in the mixing unit (T1) and the temperature in the midpoint of the pipe reactor(T2) are used as the explanatory variable, and the temperature of the fluid entering the inactivation unit (T3) as a response.
3) L4: The delay between grinding and mixing could be identified by the difference between L1 and L2, but it is estimated as a double-check for the whole procedure. In this estimation, the flowrates F1 and F2 are used as explanatory variables, while P1 and F3 are used as responses.

In this case, the real delays are unknown.

5. Results

5.1. Comparison of methods based on synthetic dataset

The methods are assessed on synthetic data by comparing estimated to true time lags; the employed metric is the Euclidean norm of the normalised variable time delay estimation error vector as reported in equation (18),

$$D_{method} = \left\| \frac{\boldsymbol{d}_{method} - \boldsymbol{d}_{true}}{\Delta_u - \Delta_l} \right\|, \tag{18}$$

where the method subscript is used to differentiate among the different methods, and Δ_l, Δ_u are, respectively, the lower and upper boundary vectors for the possible lags. The computation time for each estimation is also recorded and used as an additional assessment tool. The tuning of metaparameters is sometimes important for the stability and the performance of some of the methods. While this is less crucial for measures of depencence – and some do not have metaparameters to tune, like the pearson correlation or distance correlation – it is quite important for the other two classes of methods. All metaparameters where optimised for maximum performance, measured by root mean square error, in the most challenging scenario (low size, high complexity). These metaparameters were also verified on other sizes and complexities to assure that the performance was not affected. In the case of the metaparameters of the optimising algorithms, the base metaparameters where used; the one proposed by the authors of the original works. The overall ranking of methods across all synthetic data sets is presented in Table 4. From this table, it is clear that the performance varies substantially between methods. Distance correlation and MIC perform well on most data sets, while the VTR-based methods generally have poor performance and among the highest computation times. The remaining methods work well for some configurations but not all.

In order to understand how variable time delay estimation respond to different data set characteristics, an analysis of variance (ANOVA) was performed on D_{method} with the factors *Method*, *Size*, *Complexity*, *AR order* and *AR strength*. The model included main effects, two- and three-factor interaction, and the ANOVA table is given in Table 5. Note that most effects are statistically significant (p-value<0.001) due to the high number of residual degrees of freedom, even if the effect sizes (represented by explained variances) are very small. Focusing on the effect sizes, it is clear that factors *Method*, *Size* and *Complexity* are the ones that affect the results the most. Interaction plots for comparing factor levels of these three factors are given in Fig. 4.

Table 4
Overall results ranked by correctness: % of hits (percentage of correctly identified lags, i.e., $D_{method} = 0$). The table also shows the average D_{method} and average computational time.

Method	% of Hits	Avg. D_{method}	Std D_{method}	Avg. Computing Time (s)
dCorr	92	0,092	0,237	615,08
MIC	90	0,085	0,189	294,46
GA-MI	83	0,138	0,257	52,07
N-Norm	82	0,154	0,297	300,58
MI	80	0,158	0,243	45,50
RF	74	0,200	0,291	46,46
KPLS	71	0,244	0,310	529,37
ρ	64	0,298	0,354	0,08
τ	61	0,305	0,358	13,68
r	59	0,359	0,405	0,02
PLS-Beta	58	0,337	0,372	2,28
PLS-SR	49	0,327	0,366	2,31
VTR-LS	29	0,569	0,294	153,73
VTR-GMM	11	0,804	0,330	830,37

The results from the ANOVA are largely as expected, as some methods are known to be better suited to tackle linear problems while others are designed to model nonlinear problems. Some of the methods linked to variable importance in regression (§3.2), namely PLS-Beta, KPLS and RF, are the best performing when the true model complexity is linear, even for small sample sizes, as can be seen in the top subplot of Fig. 4. Most methods, except the VTR variants, still perform adequately at higher complexity levels as long as sample size is high (see middle and bottom subplot of Fig. 4). In the bottom subplot of Fig. 4 it is possible to see that while no methods would work perfectly at low Size and highly nonlinear Complexity there is a grouping on how the methods perform. Methods that are derived from information theory or distance correlation (i.e., MIC, dCorr, GA-MI, N-Norm, and MI) seem to perform better than the other methods, as long as sample size is high. Conversely a different group of methods have overall lower performance (i.e., higher Distance) across all sizes (e.g., ρ, τ, PLS-SR, PLS-Beta, r).

There is a small but statistically significant interaction effect between *AR Order*, *Size*, and *Complexity* in Table 5. Further investigation of this effect shows that *AR Order* has an effect when *Complexity* is “linear” and *Size* is “100". In this particular scenario, most methods have worse performance when *AR Order* is 2 (see Figure S5 in the supplementary material). *AR strength* had a negligible effect on the results and it therefore not shown.

Computation time is impacted only by method and size as that is the main factor that impacts the number of calculations performed. Fig. 5 shows the average computational time for each method and Size level.

In most cases, the time dependence is linear in the number of observations and sensors to align, however, some exceptions exist. The KPLS using the radial basis function and distance correlation are quadratic in the number of observations as they require the calculation of pairwise distances. The MI estimator and GA-MI methods are quadratic in the number of sensors to align. While computational time can be an important parameter in the choice of method for variable time delay estimation, it seldom is a strict limiting factor since the estimation is generally performed post-mortem. This might be different in the case of dynamic time delay because of the additional moving window parameter to tune. It might be important to note that, especially for measures of dependence-based methods (§3.1), the type of nonlinearity does also impact the performance as some methods perform better for different types of relationships. An example would be a sinusoidal relationship between **X** and **y**. With that said, the presented are among

Table 5
ANOVA table for assessment of how the estimation methods and data characteristics affect the results.The explained variance (each effect's sum-of-squares relative to total sum-of-squares) is used as a measure of effect size. Rows in bold have explained variances above one percent, while in italics have explained variance below one percent.

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob > F	Explained Variance
Size	27,82	2	13,91	413,16	<0.001	13,77
Complexity	28,55	2	14,27	424,02	<0.001	14,13
AROrder	*0,75*	1	*0,75*	*22,18*	<0.001	*0,37*
ARStr	*0*	1	*0*	*0,08*	*0,78*	*0*
Method	**54,14**	**13**	**4,16**	**123,71**	**<0.001**	**26,79**
Size*Complexity	**2,07**	**4**	**0,52**	**15,38**	**<0.001**	**1,03**
*Size*AROrder*	*1,55*	2	*0,78*	*23,09*	<0.001	*0,77*
*Size*ARStr*	*0,66*	2	*0,33*	*9,86*	<0.001	*0,33*
Size*Method	**3,98**	**26**	**0,15**	**4,55**	**<0.001**	**1,97**
*Complexity*AROrder*	*0,76*	2	*0,38*	*11,36*	<0.001	*0,38*
*Complexity*ARStr*	*0,52*	2	*0,26*	*7,75*	<0.001	*0,26*
Complexity*Method	**17,36**	**26**	**0,67**	**19,83**	**<0.001**	**8,59**
*AROrder*ARStr*	*0*	1	*0*	*0,14*	*0,71*	*0*
*AROrder*Method*	*1,03*	13	*0,08*	*2,36*	<0.001	*0,51*
*ARStr*Method*	*1,39*	13	*0,11*	*3,18*	<0.001	*0,69*
Size*Complexity*AROrder	**2,07**	**4**	**0,52**	**15,37**	**<0.001**	**1,02**
*Size*Complexity*ARStr*	*1,76*	4	*0,44*	*13,07*	<0.001	*0,87*
Size*Complexity*Method	**8,68**	**52**	**0,17**	**4,96**	**<0.001**	**4,3**
*Size*AROrder*ARStr*	*0,8*	2	*0,4*	*11,88*	<0.001	*0,4*
*Size*AROrder*Method*	*0,9*	26	*0,03*	*1,03*	*0,42*	*0,45*
*Size*ARStr*Method*	*0,97*	26	*0,04*	*1,11*	*0,32*	*0,48*
*Complexity*AROrder*ARStr*	*0,91*	2	*0,45*	*13,47*	<0.001	*0,45*
*Complexity*AROrder*Method*	*1,47*	26	*0,06*	*1,68*	*0,02*	*0,73*
*Complexity*ARStr*Method*	*1,57*	26	*0,06*	*1,79*	*0,01*	*0,78*
*AROrder*DynStr*Method*	*1,25*	13	*0,1*	*2,85*	<0.001	*0,62*
Error	41,07	1220	0,03			20,33
Total	202,06	1511	0			100

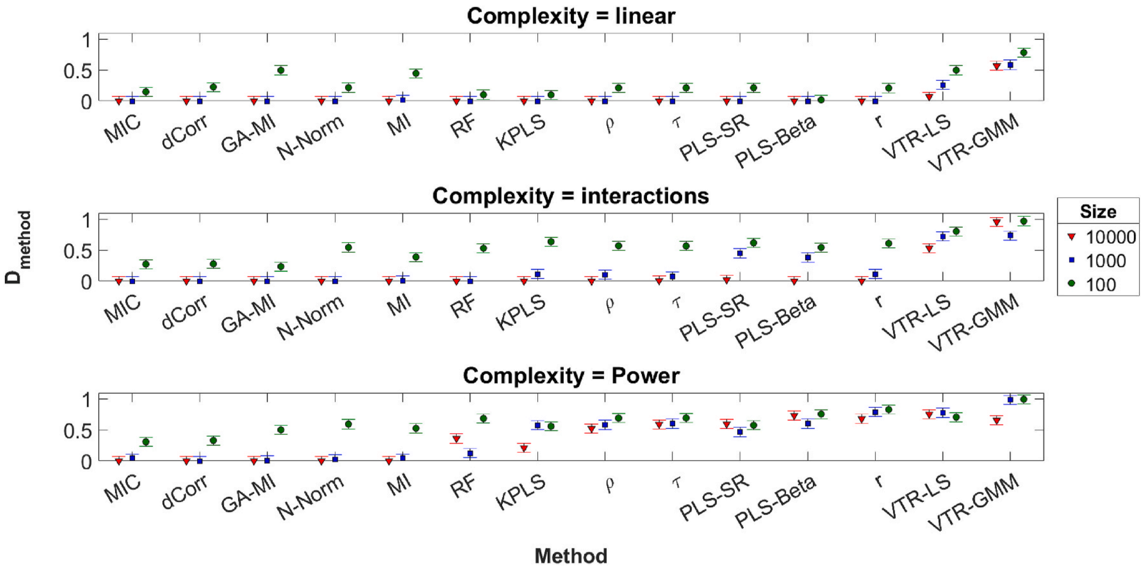


Fig. 4. Comparison of levels for all combination of the factors Method, Size, and Complexity. The dots represent group means while the error bars represent the 95% Least Significant Difference intervals.

the most common types of relationship in the process industry; for further reading on the topic, see, e.g., §4.3 in Chatterjee [61].

5.2. Application of methods on the Kamyr dataset

The methods are also applied in a semi-synthetic way to an industrial dataset to complement the application to the synthetic dataset; in this occasion, the lags are known, but the data comes from a real industrial process, as introduced in §4.2. A resampling procedure was used to evaluate the uncertainties of the estimated lags: each method was applied to 50 subsets of the dataset, sampled randomly without replacement from the augmented time series. Each subset contained 125 samples. Estimates that had an interquartile range above 3 were defined as "non-estimable". In Table 6 results are reported as the estimated delay for all analysed variables. Variable names and numbers are reported as written in Table17.1 of [94].

From Table 6, it is possible to notice some things. First,we see that many methods give similar results regarding the number of "Hits" and "Uncertain" estimations. Inspection of the PLS model reveals that the relationships between process variables and the response are mainly linear, which means that this data set resembles the "low complexity" scenario of the synthetic data. It is therefore expected that most methods work well, as seen from Fig. 4. Second, the uncertain estimation are mostly from the same variables across most methods, suggesting that

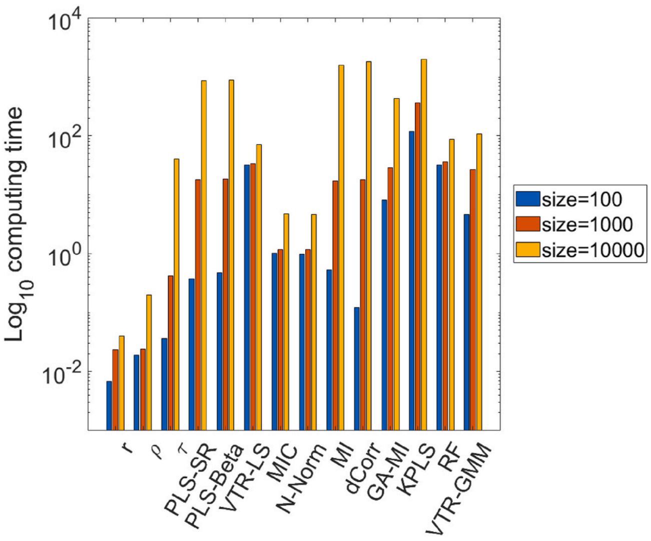


Fig. 5. Average computational time for the variable time delay estimation using different methods. Different bar colours represent different sample sizes: blue is 100, orange is 1000, and yellow is 1000. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

these variables have very weak or no relationship to the Kappa number being analysed. Third, the methods with more meta-parameters to tune have worse performance in this semi-synthetic approach compared to the fully synthetic one, probably due to the more complex meta-parameter selection; in the synthetic example, each group of three variables share trends and distribution, while in this case, most variables are different. While it is possible to optimise the models at the heart of those methods better, it is time-consuming and very delicate. Methods with no impacting meta parameters are generally better performing; examples would be dCorr, with zero meta parameters, or MIC, whose meta parameter can be safely left at the standard value [96]. Lastly, we notice that most of the well-performing methods estimate a lag of 15 for the variable *3.blcm.* This might suggest that there is a lag for this variable in the original data set, which was not corrected for prior to our analyses.

5.3. Application of methods on industrial bioprocess dataset

This procedure will refer to the variable delay estimation steps introduced in section §2. The aim is to estimate the lags between unit operations, as shown in Fig. 3, i.e., the values of L1, L2, L3 and L4, using the variables listed in Table 3.

The inactivation step was chosen as the fixed zero point, t_0, as the reaction stops there with no further chemical changes to the end-product qualities. Five minutes were chosen as the time step, following the process operators' advice and manual data inspection, as the expected residence time of the mixing vessel is around 5 min. Since the product sampling point is right after the inactivation, this choice of time resolution means that the lag between inactivation and product sampling is negligible. Furthermore, it would let us accommodate for some minor errors in the reported sampling time, which are expected as it has been done manually. The data were down-sampled using the median of 5-min periods. Also, periods when the process was not running in normal operation conditions were removed.

The dataset comprises 390 observations of the end-product qualities and process variables described in Table 3. For methods capable of processing multivariate data, the full NIR spectrum from the inline NIR sensor is used (141 wavelengths), while the first component of a PCA model built on the NIR spectra, which describes ~77% of the total variability, is used for the other methods.

The reported methods are Distance correlation, Maximal Information Coefficient, Kernel PLS and Random Forest regression. These methods are chosen as they are the two best performing for the measure of dependence (see §3.1) and variable importance in regression (see §3.2) types according to Table 4. The results for the estimation for all other methods are reported in the supplementary materials (Figures S6-S9). For Distance Correlation and Maximal Information Coefficient, the values shown are the average of 5000 estimations on subsamples of 100 random observations taken without repetition. In the KPLS and Random Forests case, the shown values are the cross-validated loss in predictive power (as explained in §3.2.2 and §3.2.3) on models trained on all 390 observations with 10-fold cross-validation. Even if autoregressive order proved to have a minor impact on the variable time delay estimation, the partial autocorrelation plot was inspected for all process variables in order to assess their autoregressive order; most variables present an order one autoregressive behaviour, while temperatures and F03 have a second order autoregressive behaviour, presumably due to the effect of

Table 6
Variable time delay estimation results for the Kamyr dataset. The #Hits row shows how many times the correct lag (11) was identified for each method, and the Uncertain row lists how many times the corresponding had an uncertainty – measured by the interquartile range of the bootstrap variable time delay estimation – above a certain threshold. The = signifies the variable where the interquartile range was above 3.

	τ	dCorr	N-Norm	PLS - Beta	MI	MIC	ρ	PLS - SR	r	RF	GA-MI	KPLS	VTR - LS	VTR - GMM
2.chip4	11	11	11	11	11	11	11	11	11	11	11	18	18	=
3.blcm	15	15	15	10	15	15	15	15	15	=	15	9	9	=
4.blfw	9	=	=	=	=	=	9	=	=	=	=	12	20	=
5.chip4	12	12	12	11	12	12	12	12	12	12	=	2	11	=
6. uxt2	=	=	11	=	11	=	=	=	=	=	=	4	12	=
7. lxt2	12	11	12	6	11	11	12	12	12	11	=	2	20	=
8. ucza3	11	11	11	11	11	11	11	11	11	11	=	11	0	=
9. wlfl4	11	11	11	11	11	11	11	11	11	11	11	20	17	=
11. aawd4	11	11	11	10	=	=	10	10	10	=	=	6	1	=
12. chmo4	12	12	12	13	12	12	12	12	12	14	12	11	12	=
13. bsfl4	11	11	11	11	11	11	11	11	11	11	11	21	20	=
14. lht3	5	6	=	=	11	10	5	6	6	10	=	15	9	=
15. uht3	=	=	=	4	=	=	=	5	=	=	=	20	20	=
16. cmfr4	11	11	11	11	11	11	11	11	11	11	11	18	20	=
17. tfflw	11	=	=	8	=	=	12	=	=	=	=	14	1	=
18. xflw2	7	7	8	=	13	=	7	8	8	=	=	17	0	=
19. f18f0	=	=	=	=	=	=	=	=	=	=	=	1	20	=
20. stfl3	12	12	12	11	12	11	12	12	12	12	12	1	0	=
21. tct4	11	11	11	11	=	11	11	11	11	=	=	2	18	=
#Hits	8	8	8	8	8	8	6	6	6	6	4	2	1	0
Uncertain	3	5	5	5	6	7	3	4	5	9	12	0	0	19

the control system.

Fig. 6 shows Variable time delay estimation performed on the industrial data with different. Methods, the black dotted line represents the point with the highest correlation/performance loss., while the green areas represent areas of uncertainty in the selection e.g., areas with the same dependence up to the second significant digit. In Fig. 6 it is shown that tested methods do not generally agree on determining the lags between the grinding and the end-product quality, estimating it between 45 and 55 min. By accounting for uncertainties – shown as light green areas – it is possible to see (Fig. 6a) that three of four are concordant. For L2 and L3, there is an agreement only if we consider the uncertainty of the estimation, e.g., the light green areas overlap for most methods, while it is not the case for the dark green area. On average, distance correlation seems to not be in accordance to the other, it might be because it incorporates the whole spectra information instead of the first principal component. In the case of measures of dependence, the uncertainty area is due to a high degree of overlap of the distributions of the measured strength of dependence in the subsampling procedure (see Fig. 7 for an example on L1). In the case of variable importance in regression methods, on the other hand, this uncertainty could be due to changes in the response that comes with the variation of meta-parameters around the optimal values – or the variation of random seed in the case of random forest regression. A further discussion point from Fig. 6 is that no methods accurately estimate L4. The last delay, shown as a red dotted line would be expected to be at the difference between L1 and L2; this could be used for "closing the loop" and validating the variable time delay estimation internally to the exercise. This misalignment is probably due to the fact that the selected period for the variable time delay estimation is small compared to the rate of change of the analysed variables, a problem of resolution. A change in raw materials quality (L1) or temperature (L2, L3) is observable in the 5-min window chosen for the estimation and has a rate of change similar to those in the variables used for the alignment. On the other hand, a change in flowrate (L4) is slow compared to the analysis period, and its rate of change is big compared to the rate of change of the pressures used for the alignment. The effects of the mixing vessel control system are added to this misalignment, which further complicates the estimation [97].

Of the measures of dependence-based methods, Distance Correlation seems to be able to identify the strength of the relationship between the multiple variables considered slightly better than MIC, especially in step L1. Fig. 7a shows the near overlap of the distributions of the estimated strength of dependence for L1 by MIC; the overlap is much smaller for Distance Correlation, as seen in Fig. 7b. The overlap is present in the other analysis steps but much less accentuated.

This different behaviour in step L1 is due to the multivariate nature of the variable time delay estimation here. When complex data types – such as spectra – and multiple targets for the alignment are present, methods that can account for the multivariate nature of the data are better suited for the time delay estimation. As mentioned above, some methods give an uncertainty area in which it is difficult to assess the strength of the dependence; these areas of uncertainty can also be used to make an informed decision on a suitable averaging period for further analysis. For example, in this case, the uncertainty among methods, and also in the same method could suggest that perhaps a larger averaging window could be beneficial. Using an averaging horizon bigger than this area would most probably dampen the variation, masking the information present at that time resolution, while using an appropriate period could help remove the need for dynamic models in the process, or at least simplify it, decreasing the order of the dynamic modelling.

5.4. General discussion

For the semi-synthetic data and the industry case, we applied

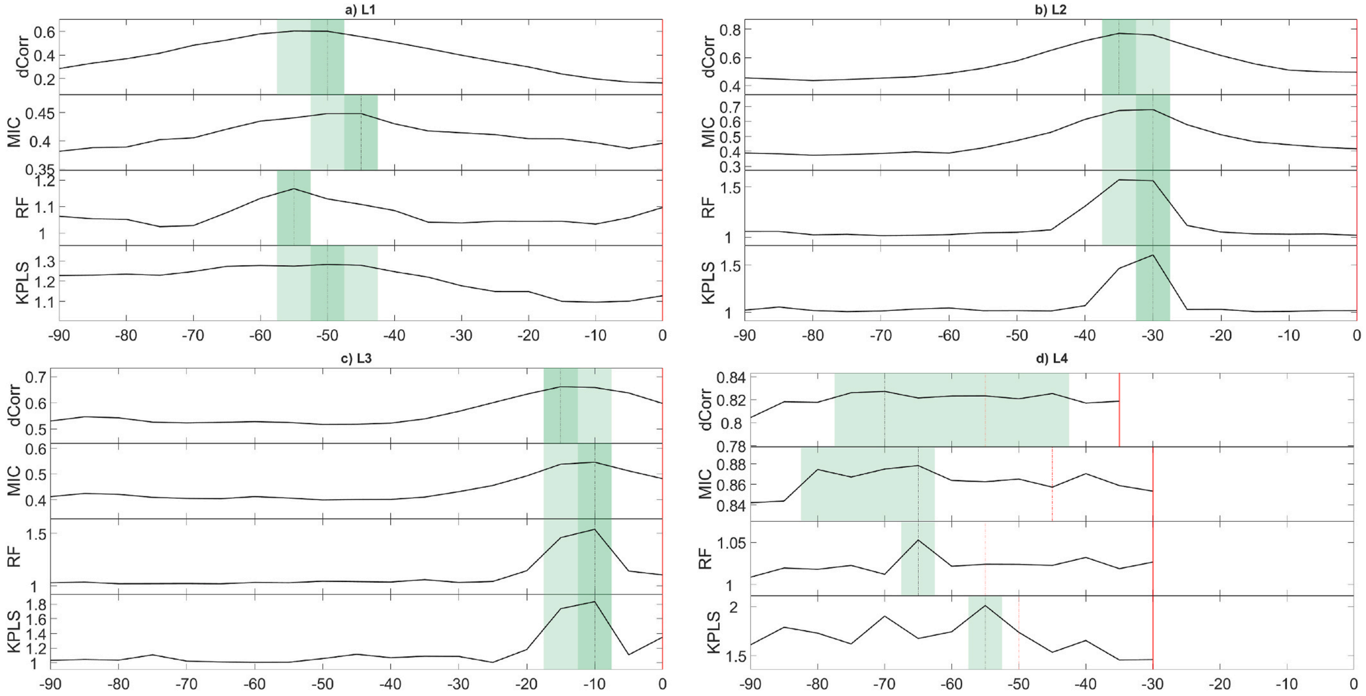


Fig. 6. Variable time delay estimation performed on the industrial data with different methods. Different subplots show different steps of the estimation exercise. The black line represents the measure of dependence (in rows 1 and 2), or the loss in predictive performance when removing each lag from the predictors (in rows 3 and 4), and the black dotted line represents the point with the highest correlation/performance loss. Furthermore, the green areas represent areas with "similar results", e.g., areas with the same dependence up to the second significant digit, and the darker green area highlights the interval with the highest correlation/performance loss, The red line identifies the starting point for the alignment; in L1, L2 and L3, it is zero, while for L4, it is the lag selected for L2. The dotted red line in c) shows the expected delay based on the difference between L1 and L2. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

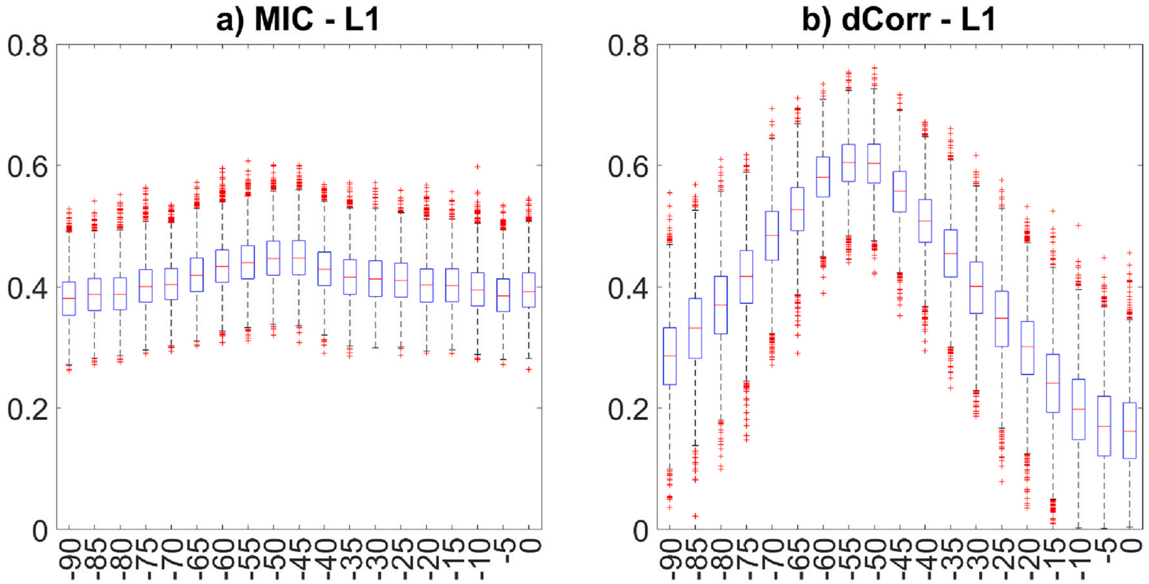


Fig. 7. Boxplot of the strength of all dependences evaluated with subsampling for two measures of dependence for step L1. a) MIC b) dCorr.

resampling techniques, either to assess the uncertainty of the lag estimates or to have better estimate of the employed indices. This practice is usually advisable, especially in cases with many explanatory variables and when the relationships between explanatory variables and the target are not known. If the lag estimates are uncertain, a solution could be to include more than one lag for the same variable. An uncertain lag estimate might also indicate that the variable have little predictive power and can probably be omitted from a soft sensor model. Additionally, it is advisable to use a panel of different methods to compare different variable time delay estimation strategies and use domain knowledge to discriminate the results.

If the sole goal of the analysis is to obtain a good prediction, it might be more efficient to directly use a machine learning method for time series forecasting on the augmented (i.e. lagged) matrix. These methods, such as long short-term memory (LSTM) cells [98], show state-of-the-art prediction performances. The variable time delay estimation described in this work is better suited in case the aim is different, e.g., knowledge discovery, a more efficient process control, a more precise fault understanding and support the development of digital twins of production processes. In most of these cases, the variable time delay estimation would be a preprocessing step and more in-depth analysis would be required afterwards.

6. Conclusion

In this paper, we have compared 14 different methods for variable time delay estimation. The methods are compared on synthetic datasets, on a semi-synthetic dataset, and on a bioprocess industry example. For the synthetic data, we explored a range of different scenarios with varying autoregressive behaviour, complexities of the relationship between variables, and sample sizes.

Our results show that most methods perform well when the relationship between variables is not too complex, and sample size is high. Methods linked to variable importance in regression (i.e., based on PLS or random forest regression) perform very well even for small samples sizes if the relationship between variables is linear. In the more demanding scenarios, methods derived from information theory outperforms the others.

Our general recommendation is to apply *Distance Correlation* (*dCorr*), *Maximal Information Coefficient* (*MIC*), *Mutual Information* (*MI*), *N-norm*, or a combination of several of these, since they work well in most scenarios, and they are easy to implement with few parameters to tune. If the sample size is small, *dCorr* and *MIC* are preferred over the other methods.

Funding

This work was funded by The Research Council of Norway through the projects SFI Digital Food Quality (project number 309259) and by the Norwegian Agricultural Food Research Foundation through the project Precision Food Production (project number 314111).

CRediT authorship contribution statement

Marco Cattaldo: Data curation, Investigation, Methodology, Software, Writing – original draft. **Alberto Ferrer:** Formal analysis, Supervision, Writing – review & editing. **Ingrid Måge:** Conceptualization, Formal analysis, Supervision, Writing – review & editing.

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.

Data availability

The authors do not have permission to share the industrial data. Code for generating the simulated data can be found at: https://github.com/CTTMRC/VTDE-in-continuous-industrial-processes, and the Kamyr dataset can be found at: https://openmv.net/info/kamyr-digester

Acknowledgements

We thank Bioco and Norilia for providing production data.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2024.105082.

References

[1] M. Bertolini, D. Mezzogori, M. Neroni, F. Zammori, Machine Learning for industrial applications: a comprehensive literature review, Expert Syst. Appl. 175 (2021) 114820, https://doi.org/10.1016/j.eswa.2021.114820.

[2] S.J. Qin, L.H. Chiang, Advances and opportunities in machine learning for process data analytics, Comput. Chem. Eng. 126 (2019) 465–473, https://doi.org/10.1016/j.compchemeng.2019.04.003.

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

[4] C. Li, Y. Chen, Y. Shang, A review of industrial big data for decision making in intelligent manufacturing, Engineering Science and Technology, Int. J. 29 (2022) 101021, https://doi.org/10.1016/J.JESTCH.2021.06.001.

[5] J.P. Usuga Cadavid, S. Lamouri, B. Grabot, R. Pellerin, A. Fortin, Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0, J. Intell. Manuf. 31 (2020) 1531–1558, https://doi.org/10.1007/s10845-019-01531-7.

[6] J. Leukel, J. González, M. Riekert, Adoption of machine learning technology for failure prediction in industrial maintenance: a systematic review, J. Manuf. Syst. 61 (2021) 87–96, https://doi.org/10.1016/J.JMSY.2021.08.012.

[7] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, J. Barbosa, Machine learning and reasoning for predictive maintenance in Industry 4.0: current status and challenges, Comput. Ind. 123 (2020) 103298, https://doi.org/10.1016/J.COMPIND.2020.103298.

[8] D.A.C. Narciso, F.G. Martins, Application of machine learning tools for energy efficiency in industry: a review, Energy Rep. 6 (2020) 1181–1199, https://doi.org/10.1016/J.EGYR.2020.04.035.

[9] Z. Kang, C. Catal, B. Tekinerdogan, Machine learning applications in production lines: a systematic literature review, Comput. Ind. Eng. 149 (2020) 106773, https://doi.org/10.1016/J.CIE.2020.106773.

[10] F. Destro, S. García Muñoz, F. Bezzo, M. Barolo, Powder composition monitoring in continuous pharmaceutical solid-dosage form manufacturing using state estimation – proof of concept, Int. J. Pharm. 605 (2021) 120808, https://doi.org/10.1016/J.IJPHARM.2021.120808.

[11] P. Facco, F. Doplicher, F. Bezzo, M. Barolo, Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerisation process, J. Process Control 19 (2009) 520–529, https://doi.org/10.1016/j.jprocont.2008.05.002.

[12] L. Fortuna, S. Graziani, M.G. Xibilia, Soft sensors for product quality monitoring in debutanizer distillation columns, Control Eng. Pract. 13 (2005) 499–508, https://doi.org/10.1016/J.CONENGPRAC.2004.04.013.

[13] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Comput. Chem. Eng. 33 (2009) 795–814, https://doi.org/10.1016/J.COMPCHEMENG.2008.12.012.

[14] M.T. Tham, G.A. Montague, A. Julian Morris, P.A. Lant, Soft-sensors for process estimation and inferential control, J. Process Control 1 (1991) 3–14, https://doi.org/10.1016/0959-1524(91)87002-F.

[15] C.C. Pantelides, J.G. Renfro, The online use of first-principles models in process operations: review, current status and future needs, Comput. Chem. Eng. 51 (2013) 136–148, https://doi.org/10.1016/J.COMPCHEMENG.2012.07.008.

[16] S. Zendehboudi, N. Rezaei, A. Lohi, Applications of hybrid models in chemical, petroleum, and energy systems: a systematic review, Appl. Energy 228 (2018) 2539–2566, https://doi.org/10.1016/J.APENERGY.2018.06.051.

[17] I. Ahmad, A. Ayub, M. Kano, I.I. Cheema, Gray-box soft sensors in process industry: current practice, and future prospects in era of big data, Processes 8 (2020) 243, https://doi.org/10.3390/pr8020243.

[18] J. Sansana, M.N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L.H. Chiang, M.S. Reis, Recent trends on hybrid modeling for Industry 4.0, Comput. Chem. Eng. 151 (2021) 107365, https://doi.org/10.1016/j.compchemeng.2021.107365.

[19] J. Zhu, Z. Ge, Z. Song, F. Gao, Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data, Annu. Rev. Control 46 (2018) 107–133, https://doi.org/10.1016/J.ARCONTROL.2018.09.003.

[20] P. Grzegorzewski, A. Kochanski, Data preprocessing in industrial manufacturing, studies in systems, Decis. Control 183 (2019) 27–41, https://doi.org/10.1007/978-3-030-03201-2_3/FIGURES/4.

[21] L. Yao, Z. Ge, Refining data-driven soft sensor modeling framework with variable time reconstruction, J. Process Control 87 (2020) 91–107, https://doi.org/10.1016/j.jprocont.2020.01.009.

[22] A.B. Corripio, C.A. Smith, Principles and Practice of Automatic Process Control, 1987. https://linkinghub.elsevier.com/retrieve/pii/0005109887900185.

[23] L. Yao, Z. Ge, Refining data-driven soft sensor modeling framework with variable time reconstruction, J. Process Control 87 (2020) 91–107, https://doi.org/10.1016/j.jprocont.2020.01.009.

[24] G.-C. Rota, D. Kahaner, A. Odlyzko, On the foundations of combinatorial theory. VIII. Finite operator calculus, J. Math. Anal. Appl. 42 (1973) 684–760, https://doi.org/10.1016/0022-247X(73)90172-8.

[25] T.J. Rato, M.S. Reis, Multiresolution soft sensors: a new class of model structures for handling multiresolution data, Ind. Eng. Chem. Res. 56 (2017) 3640–3654, https://doi.org/10.1021/acs.iecr.6b04349.

[26] M.S. Reis, Multiscale and multi-granularity process analytics: a review, Processes 7 (2019) 61, https://doi.org/10.3390/pr7020061.

[27] T. Offermans, E. Szymańska, G.H. van Kolllenburg, L.M.C. Buydens, J.J. Jansen, Automatically optimising dynamic synchronisation of individual industrial process variables for statistical modelling, Comput. Chem. Eng. 152 (2021), https://doi.org/10.1016/j.compchemeng.2021.107402.

[28] T. Offermans, E. Szymańska, L.M.C. Buydens, J.J. Jansen, Synchronising process variables in time for industrial process monitoring and control, Comput. Chem. Eng. 140 (2020), https://doi.org/10.1016/j.compchemeng.2020.106938.

[29] E. Laciar, R. Jané, D.H. Brooks, Improved alignment method for noisy high-resolution ECG and holter records using multiscale cross -correlation, IEEE Trans. Biomed. Eng. 50 (2003) 344–353, https://doi.org/10.1109/TBME.2003.808821.

[30] C. Shang, X. Gao, F. Yang, D. Huang, Novel bayesian framework for dynamic soft sensor based on support vector machine with finite impulse response, IEEE Trans. Control Syst. Technol. 22 (2014) 1550–1557, https://doi.org/10.1109/TCST.2013.2278412.

[31] D.J. Albers, G. Hripcsak, Estimation of time-delayed mutual information and bias for irregularly and sparsely sampled time-series, Chaos, Solit. Fractals 45 (2012) 853–860, https://doi.org/10.1016/j.chaos.2012.03.003.

[32] P. Chen, G. Xie, H. Liu, L. Liang, H. Gao, D. Wang, W.J. Ji, Online output estimation for multimode process with dynamic time-delay, Chinese control conference, CCC, 5742–5747, https://doi.org/10.23919/CCC50068.2020.9189279, 2020.

[33] J. Li, T. Dong, S. Zhang, X. Zhang, S.-P. Yang, Time-delay identification in dynamic processes with disturbance via correlation analysis, Control Eng. Pract. 62 (2017) 92–101, https://doi.org/10.1016/j.conengprac.2017.03.007.

[34] X. Chen, C. Zhao, Linear and nonlinear hierarchical multivariate time delay analytics for dynamic modeling and process monitoring, J. Process Control 107 (2021) 83–93, https://doi.org/10.1016/J.JPROCONT.2021.10.008.

[35] B.S. Everitt, A. Skrondal, The Cambridge Dictionary of Statistics, Cambridge University Press, Cambridge, 2010, https://doi.org/10.1017/CBO9780511779633.

[36] D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P. J. Turnbaugh, E.S. Lander, M. Mitzenmacher, P.C. Sabeti, Detecting novel associations in large data sets, Science 334 (2011) (1979) 1518–1524, https://doi.org/10.1126/science.1205438.

[37] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423, https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[38] S. Graziani, M.G. Xibilia, On the use of correlation analysis in the estimation of finite-time delay in Soft Sensors design, in: 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), IEEE, 2021, pp. 1–6, https://doi.org/10.1109/I2MTC50364.2021.9459807.

[39] G.J. Székely, M.L. Rizzo, N.K. Bakirov, Measuring and testing dependence by correlation of distances, Ann. Stat. 35 (2007) 2769–2794, https://doi.org/10.1214/009053607000000505.

[40] P. Geladi, B.R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17, https://doi.org/10.1016/0003-2670(86)80028-9.

[41] T. Rajalahti, R. Arneberg, F.S. Berven, K.M. Myhr, R.J. Ulvik, O.M. Kvalheim, Biomarker discovery in mass spectral profiles by means of selectivity ratio plot, Chemometr. Intell. Lab. Syst. 95 (2009) 35–48, https://doi.org/10.1016/j.chemolab.2008.08.004.

[42] R. Rosipal, L.J. Trejo, Kernel partial least squares regression in reproducing kernel hilbert space, J. Mach. Learn. Res. 1 (2000) 97–123, https://doi.org/10.1162/15324430260185556. CrossRef Listing of Deleted DOIs.

[43] L. Breiman, Random Forests, 2001.

[44] O. Ludwig, U. Nunes, R. Araújo, L. Schnitman, H.A. Lepikson, Applications of information theory, genetic algorithms, and neural models to predict oil flow, Commun. Nonlinear Sci. Numer. Simul. 14 (2009) 2870–2885, https://doi.org/10.1016/j.cnsns.2008.12.011.

[45] C. Croux, C. Dehon, C. Croux, C. Dehon, Influence functions of the Spearman and Kendall correlation measures, Stat. Methods Appl. 19 (2010) 497–515, https://doi.org/10.1007/s10260-010-0142-z.

[46] J. Zhang, Q. Jin, Y. Xu, Inferential estimation of polymer melt index using sequentially trained bootstrap aggregated neural networks. https://doi.org/10.1002/ceat.200500352, 2006.

[47] R. Smith, A mutual information approach to calculating nonlinearity, Stat 4 (2015) 291–303, https://doi.org/10.1002/sta4.96.

[48] C.M. Holmes, I. Nemenman, Estimation of mutual information for real-valued data with error bars and controlled bias, Phys. Rev. E 100 (2019) 1–10, https://doi.org/10.1103/PhysRevE.100.022404.

[49] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Phys. Rev. E 69 (2004) 066138, https://doi.org/10.1103/PhysRevE.69.066138.

[50] Y. Il Moon, B. Rajagopalan, U. Lall, Estimation of mutual information using kernel density estimators, Phys. Rev. E 52 (1995) 2318, https://doi.org/10.1103/PhysRevE.52.2318.

[51] B.C. Ross, Mutual information between discrete and continuous data sets. https://doi.org/10.1371/journal.pone.0087357, 2014.

[52] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (2014) 175–186, https://doi.org/10.1007/s00521-013-1368-0.

[53] F. Souza, P. Santos, R. Araújo, Variable and delay selection using neural networks and mutual information for data-driven soft sensors, in: Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010, 2010, pp. 1–8, https://doi.org/10.1109/ETFA.2010.5641329.

[54] F. Souza, R. Araújo, Variable and time-lag selection using empirical data, in: IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2011, https://doi.org/10.1109/ETFA.2011.6059083.

[55] H. Stögbauer, A. Kraskov, S.A. Astakhov, P. Grassberger, Least-dependent-component analysis based on mutual information, Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics 70 (2004) 17, https://doi.org/10.1103/PhysRevE.70.066123.

[56] A. Rényi, On measures of dependence, Acta Math. Acad. Sci. Hungar. 10 (1959) 441–451, https://doi.org/10.1007/BF02024507.

[57] E.H. Linfoot, An informational measure of correlation, Inf. Control 1 (1957) 85–89, https://doi.org/10.1016/S0019-9958(57)90116-X.

[58] Y.A. Reshef, † David, N. Reshef, P.C. Sabeti, M.M. Mitzenmacher, Equitability, Interval Estimation, and Statistical Power, n.d..

[59] Y.A. Reshef, † David, N. Reshef, H.K. Finucane, P.C. Sabeti, M.M. Mitzenmacher, Measuring Dependence Powerfully and Equitably, n.d..

[60] J.B. Kinney, G.S. Atwal, Equitability, mutual information, and the maximal information coefficient, Proc. Natl. Acad. Sci. U. S. A. 111 (2014) 3354–3359, https://doi.org/10.1073/pnas.1309933111.

[61] S. Chatterjee, A new coefficient of correlation, J. Am. Stat. Assoc. 116 (2021) 2009–2022, https://doi.org/10.1080/01621459.2020.1758115.

[62] T. Liang, Q. Zhang, X. Liu, C. Lou, X. Liu, H. Wang, Time-Frequency Maximal Information Coefficient Method and its Application to Functional Corticomuscular Coupling Index Terms-Ankle dorsiflexion, functional corticomus-cular coupling, specific frequency band, time-frequency maximal information coefficient, IEEE Trans. Neural Syst. Rehabil. Eng. 28 (2020), https://doi.org/10.1109/TNSRE.2020.3028199.

[63] H. Iuchi, M. Sugimoto, M. Tomita, MICOP: maximal information coefficient-based oscillation prediction to detect biological rhythms in proteomics data, BMC Bioinf. 19 (2018) 249, https://doi.org/10.1186/s12859-018-2257-4.

[64] G.J. Székely, M.L. Rizzo, Brownian distance covariance, 1236–1265, https://doi.org/10.1214/09-AOAS312, 2009.

[65] G.J. Székely, M.L. Rizzo, Partial distance correlation with methods for dissimilarities, Ann. Stat. 42 (2014) 2382–2412, https://doi.org/10.1214/14-AOS1255.

[66] D. Edelmann, K. Fokianos, M. Pitsillou, An updated literature review of distance correlation and its applications to time series, Int. Stat. Rev. 87 (2019) 237–262, https://doi.org/10.1111/insr.12294.

[67] E. Martínez-Gómez, M.T. Richards, D.S.P. Richards, Distance correlation methods for discovering associations in large astrophysical databases, Astrophys. J. 781 (2014), https://doi.org/10.1088/0004-637X/781/1/39.

[68] D.S. Matteson, R.S. Tsay, Independent component analysis via distance covariance, J. Am. Stat. Assoc. 112 (2017) 623–637, https://doi.org/10.1080/01621459.2016.1150851.

[69] M.L. Rizzo, G.J. Székely, DISCO analysis: a non-parametric extension of analysis of variance, Ann. Appl. Stat. 4 (2010) 1034–1055, https://doi.org/10.1214/09-AOAS245.

[70] C. Shen, C.E. Priebe, J.T. Vogelstein, From distance correlation to multiscale graph correlation, J. Am. Stat. Assoc. 115 (2020) 280–291, https://doi.org/10.1080/01621459.2018.1543125.

[71] S. Graziani, M.G. Xibilia, Design of a soft sensor for an industrial plant with unknown delay by using deep learning, in: 2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2019, pp. 1–6, https://doi.org/10.1109/I2MTC.2019.8827074, 2019-May.

[72] S. Graziani, M.G. Xibilia, Multiple correlation analysis for finite-time delay estimation in Soft Sensors design, in: Conference Record - IEEE Instrumentation and Measurement Technology Conference, 2022, https://doi.org/10.1109/I2MTC48687.2022.9806576.

[73] D. Albanese, M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello, Minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers, Bioinformatics 29 (2013) 407–408, https://doi.org/10.1093/bioinformatics/bts707.

[74] C.M. Andersen, R. Bro, Variable selection in regression-a tutorial, J. Chemom. 24 (2010) 728–737, https://doi.org/10.1002/cem.1360.

[75] G. Heinze, C. Wallisch, D. Dunkler, Variable selection – a review and recommendations for the practicing statistician, Biom. J. 60 (2018) 431–449, https://doi.org/10.1002/bimj.201700067.

[76] M.G. Xibilia, N. Gemelli, G. Consolo, Input variables selection criteria for data-driven Soft Sensors design, in: Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control, ICNSC, vol. 2017, 2017, pp. 362–367, https://doi.org/10.1109/ICNSC.2017.8000119.

[77] P. Nomikos, J.F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE J. 40 (1994) 1361–1375, https://doi.org/10.1002/aic.690400809.

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

[79] T. Mehmood, S. Sæbø, K.H. Liland, Comparison of variable selection methods in partial least squares regression, J. Chemom. 34 (2020), https://doi.org/10.1002/cem.3226.

[80] T. Mehmood, K.H. Liland, L. Snipen, S. Sæbø, A review of variable selection methods in Partial Least Squares Regression, Chemometr. Intell. Lab. Syst. 118 (2012) 62–69, https://doi.org/10.1016/j.chemolab.2012.07.010.

[81] K.E. Pilario, M. Shafiee, Y. Cao, L. Lao, S.H. Yang, A review of kernel methods for feature extraction in nonlinear process monitoring, Processes 8 (2020) 1–47, https://doi.org/10.3390/pr8010024.

[82] U. Talukdar, S.M. Hazarika, J.Q. Gan, A Kernel Partial least square based feature selection method, Pattern Recogn. 83 (2018) 91–106, https://doi.org/10.1016/j.patcog.2018.05.012.

[83] Q. Jia, Y. Zhang, Quality-related fault detection approach based on dynamic kernel partial least squares, Chem. Eng. Res. Des. 106 (2016) 242–252, https://doi.org/10.1016/j.cherd.2015.12.015.

[84] J.M. Lee, C.K. Yoo, S.W. Choi, P.A. Vanrolleghem, I.B. Lee, Nonlinear process monitoring using kernel principal component analysis, Chem. Eng. Sci. 59 (2004) 223–234, https://doi.org/10.1016/j.ces.2003.09.012.

[85] R. Vitale, O.E. de Noord, A. Ferrer, A kernel-based approach for fault diagnosis in batch processes, J. Chemom. 28 (2014) S697–S707, https://doi.org/10.1002/cem.2629.

[86] P.W.T. Krooshof, B. Üstün, G.J. Postma, L.M.C. Buydens, Visualization and recovery of the (Bio)chemical interesting variables in data analysis with support vector machine classification, Anal. Chem. 82 (2010) 7000–7007, https://doi.org/10.1021/ac101338y.

[87] G.J. Postma, P.W.T. Krooshof, L.M.C. Buydens, Opening the kernel of kernel partial least squares and support vector machines, Anal. Chim. Acta 705 (2011) 123–134, https://doi.org/10.1016/j.aca.2011.04.025.

[88] A. Fisher, C. Rudin, F. Dominici, All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously, J. Mach. Learn. Res. 20 (2019) 1–81. http://jmlr.org/papers/v20/18-760.html. (Accessed 28 July 2022).

[89] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140, https://doi.org/10.1007/BF00058655.

[90] G. Biau, E. Scornet, A random forest guided tour, Test 25 (2016) 197–227, https://doi.org/10.1007/s11749-016-0481-7.

[91] T. Hastie, J. Friedman, R. Tibshirani, The Elements of Statistical Learning, Springer, New York, New York, NY, 2001, https://doi.org/10.1007/978-0-387-21606-5.

[92] L. Yao, Z. Ge, Cooperative deep dynamic feature extraction and variable time-delay estimation for industrial quality prediction, IEEE Trans. Ind. Inf. 17 (2021) 3782–3792, https://doi.org/10.1109/TII.2020.3021047.

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

[94] L. Eriksson, T. Byrne, E. Johansson, J. Trygg, C. Vikström, Multi-and Megavariate Data Analysis Basic Principles and Applications, Umetrics Academy, 2013.

[95] B.S. Dayal, J.F. Macgregor, P.A. Taylor, R. Kildaw, S. Marcikic, Application of feedforward neural networks and partial least-squares regression for modeling kappa-number in a continuous KAMYR digester, PULP & PAPER-CANADA 95 (1994) 26–32.

[96] D.N. Reshef, Y.A. Reshef, P.C. Sabeti, M. Mitzenmacher, An empirical study of the maximal and total information coefficients and leading measures of dependence, Ann. Appl. Stat. 12 (2018) 123–155, https://doi.org/10.1214/17-AOAS1093.

[97] L.J. Li, T.T. Dong, S. Zhang, X.X. Zhang, S.-P.P. Yang, J. Li, T.T. Dong, S. Zhang, X. X. Zhang, S.-P.P. Yang, Time-delay identification in dynamic processes with disturbance via correlation analysis, Control Eng. Pract. 62 (2017) 92–101, https://doi.org/10.1016/j.conengprac.2017.03.007.

[98] G. Van Houdt, C. Mosquera, G. Nápoles, A review on the long short-term memory model, Artif. Intell. Rev. 53 (2020) 5929–5955, https://doi.org/10.1007/s10462-020-09838-1.