Analytica Chimica Acta 1362 (2025) 344167

Contents lists available at ScienceDirect

Analytica Chimica Acta

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

Locally-weighted-RoBoost-PLS: a multivariate calibration approach to simultaneously cope with non-linearities and outliers

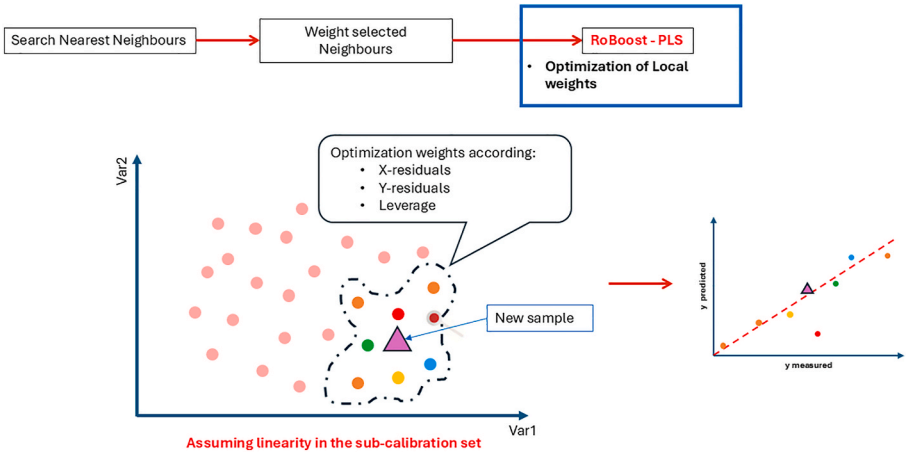
Daniele Tanzilli [a,b,1,*], Lorenzo Strani [a,1], Maxime Metz [c], Jean Michel Roger [d,e], Matthieu Lesnoff [e,f], Cyril Ruckebusch [b], Marina Cocchi [a], Raffaele Vitale [b]

[a] University of Modena and Reggio Emilia, Department of Chemical and Geological Sciences, Via Campi 103, Modena, 41125, Italy
[b] Univ. Lille, CNRS, LASIRE, Laboratoire Avancé de Spectroscopie pour les Interactions, la Réactivité et l'Environnement, F-59000 Lille, France
[c] Pellenc ST, Provence-Alpes-Côte d'Azur, Pertuis, France
[d] ITAP, Univ. Montpellier, INRAE, Institut Agro, Montpellier, France
[e] ChemHouse Research Group, Montpellier, France
[f] UMR SELMET, Univ Montpellier, CIRAD, INRA, Institut Agro, Montpellier, France

HIGHLIGHTS

- **Addresses challenges** in handling multivariate data with non-linearity and outliers.
- **Introduces LW-RoBoost-PLS**, a novel method combining local and robust modelling.
- **A multiblock extension** of the proposed LW-RoBoost-PLS is also provided.
- **ABS production**, improving real-time industrial quality prediction.

GRAPHICAL ABSTRACT



ARTICLE INFO

Handling Editor: Prof. L. Buydens

Keywords:
Partial least squares regression (PLS)
K-nearest-neighbours-locally-weighted-PLS (KNN-LW-PLS)

ABSTRACT

Background: Partial Least Squares regression (PLS) is a widely used tool for predictive modelling, particularly when dealing with multivariate datasets with dependent variables exhibiting strong collinearities. However, when relationships between variables are non-linear or atypical data points have to be coped with, PLS calibration models may face challenges. In recent years, different variants of the original PLS algorithm have been proposed to overcome these limitations. On the one hand, several robust regression methods that down-weigh outlying observations during the model training phase like RoBoost-PLS have been developed to reduce the

This article is part of a special issue entitled: CAC2024 published in Analytica Chimica Acta.

* Corresponding author. University of Modena and Reggio Emilia, Department of Chemical and Geological Sciences, Via Campi 103, Modena, 41125, Italy.
E-mail address: daniele.tanzilli@unimore.it (D. Tanzilli).

[1] Daniele Tanzilli and Lorenzo Strani have contributed equally.

https://doi.org/10.1016/j.aca.2025.344167
Received 30 January 2025; Received in revised form 28 April 2025; Accepted 8 May 2025
Available online 8 May 2025

RoBoost-PLS
Non-linearities
Outliers

detrimental effect of outliers on the performance of PLS. On the other hand, local modelling approaches, like *K*-Nearest-Neighbours-Locally-Weighted-PLS (KNN-LW-PLS), have been designed to handle non-linearities by fitting for each new incoming sample a separate linear calibration model considering only its nearest-neighbours. Unfortunately, none of these strategies can address the two aforementioned problems simultaneously. This paper introduces a novel approach named Locally-Weighted-RoBoost-PLS (LW-RoBoost-PLS), that combines the strengths of both local and robust modelling methodologies in order to deal with non-linearities while mitigating at the same time the influence of outliers.

Results: The performance of LW-RoBoost-PLS was evaluated on simulated and real industrial data (with this latter resulting from a continuous Acrylonitrile-Butadiene-Styrene ABS production process conducted at Versalis S.p.A.), both characterised by the simultaneous presence of outliers and non-linear relationships among measured variables. In the two case-studies investigated here, LW-RoBoost-PLS outperformed RoBoost-PLS and KNN-LW-PLS, achieving considerable reductions in the prediction error and prediction bias, which demonstrates that this technique permits to effectively overcome the limitations of the other approaches.

Significance: This paper describes a novel multivariate calibration approach named LW-RoBoost-PLS, which provides a solution for predictive modelling in scenarios where outliers and non-linearities co-exist. LW-RoBoost-PLS simultaneously handles non-linearities and outliers by combining local and robust modelling strategies, leading to improved prediction accuracy and reduced bias.

1. Introduction

Partial Least Squares regression (PLS) [1,2] is a widely used multivariate calibration tool for modelling linear relationships between independent (predictors or regressors) and dependent variables (responses). In contrast to classical multilinear regression, it excels in scenarios in which high-dimensional datasets exhibiting a high degree of collinearity are involved. The main advantage PLS brings regards the fact that it provides not only a model linking the aforementioned independent and dependent variables but also a full description of both in terms of latent factors, thus enabling an in-depth interpretation of the final results it yields.

In spite of all these benefits, though, PLS can suffer from severe limitations when the relationships between regressors (usually encoded in a matrix denoted as **X** of dimensions *N x V*) and responses (usually encoded in a matrix denoted as **Y** of dimensions *N x M*) are non-linear. This situation can occur quite frequently, *e.g.*, when dealing with agronomic samples collected during different harvesting campaigns [3,4], and in industry when the same plant manufactures different products through smooth formulation changes [5] or undergoes temporal drifts due to raw material or catalyst degradation [6]. If non-linearities are moderate, non-linear (*e.g.*, logarithmic) transformations of the response (s) [7] can be used to model them, otherwise, in more complex situations, non-linear implementations of PLS should be considered [8,9]. These extensions of PLS include, for example, Kernel PLS (K-PLS [10–12]) or local PLS [13–15]. K-PLS handles non-linearities by applying a specific kernel function (polynomial, gaussian, sigmoidal, *etc*.) to **X** for mapping the original regressors into a higher-dimensional feature space where a linear PLS model can be constructed. On the other hand, the main idea behind local PLS is to build for each new specimen whose responses are to be predicted an individual PLS model on a reduced subset of calibration samples that are most similar to it [13–18]. Local PLS implementations are definitely the most commonly utilized in chemometrics. Although K-PLS has shown excellent performance compared to other methods, the optimization of the kernel function remains a critical step and can become cumbersome in the presence of local nonlinearities [19]. Additionally, interpreting K-PLS models can be challenging since the relevance of the original variables is lost during the mapping process. In fact, even if recent studies have highlighted that suitable sensitivity analysis can restore the interpretability of such models by providing sensitivity vectors in the space of the real variables – which can be useful for variable selection and qualitative interpretation [20] – their coefficients are not readily interpretable [21]. Moreover, K-PLS may require long processing times when dealing with particularly large sample sizes [11]. Among local PLS implementations, *K*-Nearest-Neighbours-Locally-Weighted-PLS-Regression (KNN-LW-PLS) [22] is undoubtedly the one that has lately attracted more attention from users and practitioners. More in detail, KNN-LW-PLS trains local models exactly as outlined before, but, in addition, it weighs the samples belonging to the calibration subset according to their distance to the one to be assessed. This allows effectively capturing and describing strong non-linearities and complex patterns in data, such as the presence of distinct observation clusters, that may hamper the application of classical linear PLS to the entire dataset at hand.

Another issue that can dramatically jeopardize the predictive performance of standard PLS is the presence of outliers in the calibration data. In this regard, in the last decades, many robust versions of PLS have been developed in an attempt to downweigh outlying samples and, thus, reduce their influence in the PLS model calibration stage [23–26]. In this article, we particularly focus on a recent robust PLS implementation, RoBoost-PLS [27,28], which stands out for its ability to reduce the influence of outliers during calibration by weighting the investigated samples differently for each extracted latent factor and according to three distinct criteria: **X**-residuals, **Y**-residuals and leverage. This method has proven to be effective when it comes to dealing with outliers in both **Y** and **X**.

Although all these PLS extensions (non-linear and robust) perform well when trying to tackle the specific target problem for which they have been originally proposed, they may encounter difficulties in situations where both non-linearities and outliers coexist. Notwithstanding, to the best of our knowledge, no PLS algorithm capable of handling both these issues simultaneously has been devised yet. For this reason, we propose here a novel approach based on a rational combination of KNN-LW-PLS and RoBoost-PLS and named Locally-Weighted-RoBoost-PLS (LW-RoBoost-PLS), designed to deal with similar scenarios. Although the method still uses KNN for local weighting, the name has been simplified by dropping the 'KNN-' for ease of reading. The performance of LW-RoBoost-PLS was here evaluated on simulated and real datasets and compared with that of the two native approaches from which it originates.

2. Materials and methods

2.1. Datasets

LW-RoBoost-PLS was tested on simulated data generated by using a R software package developed by Metz et al. [29] (available at https://github.com/maxmetz/data_simulation) and on a challenging real-world dataset related to the production process of Acrylonitrile-Styrene-Butadiene (ABS). Both simulated and real-world data exhibit outliers and non-linear variable relationships, making them suitable for assessing the effectiveness of the proposed approach.

2.1.1. Simulated data

A spectral-like dataset characterized by the presence of both **X**- and **Y**-outliers and by a non-linear dependence between **X** and **Y** was simulated as detailed below.

A dataset **X** of dimensions *N* x *V* was generated multiplying a 900 x 10 matrix (say **R**) whose columns carried values drawn from a normal distribution with mean equal to 0 and standard deviation equal to 0.5 by a 10 x 1001 array (say $\mathbf{B}^T$) whose rows contained ten different pseudo-spectral profiles (see Equation (1)) – the pseudo-spectral profiles were obtained by the combination of 10 distinct Gaussian curves:

$$\mathbf{X} = \mathbf{R}\,\mathbf{B}^T \tag{1}$$

An individual dependent variable, **y**, was afterwards simulated based on the following equation to ensure a non-linear dependence with **X**:

$$\mathbf{y} = \mathbf{t}_1^2 + 15 \tag{2}$$

where $\mathbf{t}_1$ (*N* x *1*) gathered the first principal component scores resulting from the Principal Component Analysis (PCA) decomposition of **X**. Both **X** and **y** were afterwards augmented with outlying observations. 30 moderate **X**-outliers were produced as in Equation (1) but drawing the values along each column of **R** from a normal distribution with mean equal to 1 and standard deviation equal to 0.8. 40 extreme **X**-outliers were instead produced altering both **R** and $\mathbf{B}^T$, *i.e.*, drawing the values along the columns of **R** from a normal distribution with mean equal to 1.5 and standard deviation equal to 0.5, and using different pseudo-spectral profiles for $\mathbf{B}^T$. 30 **y**-outliers were, finally, obtained modifying Equation (2) as:

$$\mathbf{y}_{out} = \mathbf{t}_1 + 17 \tag{3}$$

Ultimately, Gaussian white noise was added to both **X** and **y**.

For the sake of a fair comparison among KNN-LW-PLS, RoBoost-PLS and LW-RoBoost-PLS, the entire dataset was split into a calibration set of 700 rows and a test set of 300 rows. It is important to notice here that outliers were exclusively kept in the calibration set and that they constituted approximately 15 % of it. Fig. 1 displays the simulated calibration data and their corresponding **y**-value distribution.

2.1.2. ABS data

The real dataset investigated in this article relates to a manufacturing campaign of ABS conducted from January 2020 to April 2022 in an industrial plant owned by ENI Versalis and located in Mantova (Italy). This plant operates a continuous five-stage process resulting in nine different grades of ABS as detailed in Refs. [30–32]. A smooth transition from one grade to another is operated without stopping the production. Such a process is monitored by means of 118 Process Sensors (PS) that measure temperatures, flow rates, pressures, and motor speeds, while four Matrix Fourier Transform-Near InfraRed (FT-NIR) spectrometers (Bruker Optics, Milan, Italy) are employed at four different plant locations to acquire near-infrared spectra of raw materials, intermediates and final products. These data constitute a good benchmark for testing LW-RoBoost-PLS given the presence of both outliers and non-linearities induced by the continuously varying properties of the different ABS grades manufactured.

The real-time acquisition of the aforementioned measurements yielded nine distinct data blocks: five containing the values registered through the process sensors during the different production stages, and four carrying the collected FT-NIR spectral profiles. The predictive model distinguishes between process phases to account for the physical and chemical transformations occurring at different production stages. Sensors are strategically placed to capture key transitions, such as the dissolution of butadiene in styrene, reagent recovery, and reaction progression. This phase-based approach is particularly valuable because it enables early estimation of the final product's quality. This data block division accounts for the physical and chemical transformations occurring at the different production phases and enables the early estimation of the final product's quality. Sensors, indeed, are strategically placed to capture key process transitions, such as the dissolution of butadiene in styrene, the reagent recovery, and the reaction progression.

All the recordings in these blocks were synchronised and concatenated row-wise according to the structure of the process pipeline: synchronization involved the recording alignment based on a time stamp provided by ENI Versalis and ensured that each row of the final data matrix carried information about the same section of the production flow. Hence, this data matrix was used to build KNN-LW-PLS, RoBoost PLS and LW-RoBoost PLS regression models for the prediction of a single quality parameter, referred to as Property 1 for confidentiality reasons. Reference values for Property 1 were retrieved through offline analyses three times per day. These values, expressed in grams, give insights into the physical features of the copolymer.

In the present study, the entire available dataset was split into calibration and test sets, spanning the first two years of production (2020–2021, 1851 measurement points) and the remaining

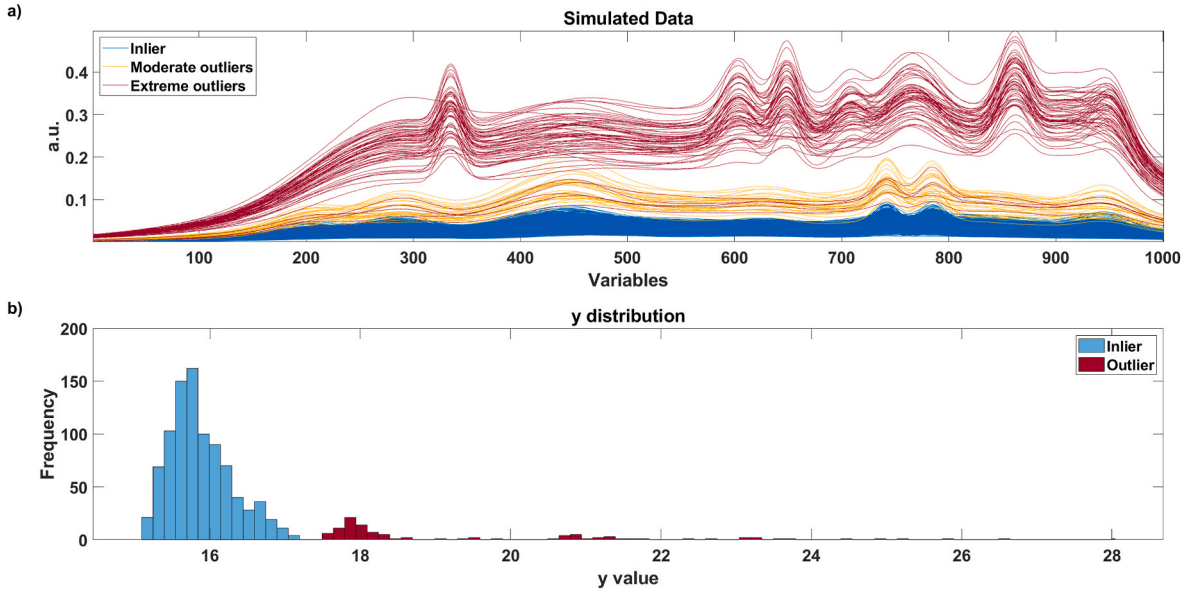


Fig. 1. a) Simulated calibration spectra and b) their corresponding **y**-value distribution.

manufacturing period (2022, 344 measurement points), respectively. The test set was subjected to a data cleaning step aimed at minimising the presence of outliers in order to ensure a fair comparison of results between the different approaches.

2.2. Methods

This section provides an overview of PLS, KNN-LW-PLS and RoBoost-PLS as well as a comprehensive description of the algorithmic scheme underlying LW-RoBoost-PLS and of the double cross-validation procedure designed for the optimization of its tuneable parameters.

2.2.1. Partial Least Squares regression (PLS)

Among the available PLS algorithms, we here refer to Nonlinear Iterative Partial Least Squares (NIPALS, on which also RoBoost-PLS is based), originally proposed by Herman Wold in the 1970s [2] and later adapted and modified by Svante Wold and Harald Martens [1]. NIPALS is an iterative approach that calculates PLS components one at a time by sequentially deflating from the data at hand the variability accounted for by the one estimated in the previous computational step.

Algorithm 1: NIPALS

Given a predictor matrix **X** (*N* x *V*) and a response matrix **Y** (*N* x *M*), both centred, NIPALS proceeds as follows:

1. Initialize **Y**-scores, $\mathbf{u}_a$, as a column of **Y**
2. Calculate **X**-weights for the *a*-th latent variable

$$\mathbf{w}_a = \mathbf{X}^T \mathbf{u}_a \left(\mathbf{u}_a^T \mathbf{u}_a\right)^{-1}$$

3. Calculate **X**-scores

$$\mathbf{t}_a = \mathbf{X}\,\mathbf{w}_a$$

4. Define **Y**-loadings

$$\mathbf{q}_a = \mathbf{Y}_a^T\,\mathbf{t}_a \left(\mathbf{t}_a^T\,\mathbf{t}_a\right)^{-1}$$

5. Update $\mathbf{u}_a$

$$\mathbf{u}_a = \mathbf{Y}\,\mathbf{q}_a \left(\mathbf{q}_a^T\,\mathbf{q}_a\right)^{-1}$$

6. Repeat steps 2 to 5 until convergence (*i.e.*, until the difference between consecutive estimations of $\mathbf{u}_a$ is found to be below a user-defined threshold)
7. Calculate **X**-loadings

$$\mathbf{p}_a = \mathbf{X}^T\,\mathbf{t}_a \left(\mathbf{t}_a^T\,\mathbf{t}_a\right)^{-1}$$

8. Deflate **X** and **Y**

$$\mathbf{X} = \mathbf{X} - \mathbf{t}_a\,\mathbf{p}_a^T$$

$$\mathbf{Y} = \mathbf{Y} - \mathbf{t}_a\,\mathbf{q}_a^T$$

9. Repeat steps 2 to 8 until all the required latent variables (*A*) are extracted
10. Calculate regression coefficients

$$\mathbf{B} = \mathbf{W}\left(\mathbf{P}^T\mathbf{W}\right)^{-1}\mathbf{Q}^T$$

The prediction of the response associated with a new measured observation $\mathbf{x}_{new}$ is then obtained as:

$$\hat{\mathbf{y}} = (\mathbf{x}_{new} - \overline{\mathbf{x}})\,\mathbf{B} + \overline{\mathbf{y}}$$

with $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ being the vectors containing the column means of **X** and **Y**, respectively.

2.2.2. K-Nearest-Neighbours-Locally-Weighted-PLS (KNN-LW-PLS)

The essential idea behind local PLS approaches is i) to select, among all the calibration samples available, a calibration subset made up of those that are most similar to each new specimen whose responses are to be estimated and ii) to calibrate a linear PLS regression model on such a reduced subset of observations for prediction purposes. Even if computationally demanding (notice that this way an individual PLS model is constructed for every new incoming test sample), this strategy permits to readily handle complex non-linearities linking predictors and responses.

Among the various local PLS implementations described in literature [15,17,33,34], in the present study we focus on the one proposed by Lesnoff et al., KNN-LW-PLS [22]. However, for the sake of simplicity and consistency, the scheme illustrated in Algorithm 2 is based on NIPALS rather than on the Dayal-MacGregor algorithm [35] as originally reported in Ref. [22]. KNN-LW-PLS consists of three fundamental steps. First, the distance between a test sample and each training sample is calculated and a user-defined number of nearest neighbours, *K*, of such a test sample is selected. Then, these nearest neighbours are weighted according to their similarity (distance) to the test sample to be assessed using the following weighting function:

$$w_{\text{local},n} = exp\left(-\frac{d_n^*}{h\,\sigma(\mathbf{d}^*)}\right) \tag{4}$$

where $\mathbf{d}^*$ is a vector containing all the max-normalized distance values calculated, d_n^* is the max-normalized distance computed for the *n*-th neighbour, σ denotes the standard deviation operator and *h* represents a parameter that influences the shape of the weighting function. The higher *h*, the lesser d_n^* affects the weights. For infinite values of *h*, every selected calibration sample has the same weight.

In a nutshell, once the calibration subset has been identified, a weighted-PLS model is constructed as outlined below:

Algorithm 2: KNN-LW-PLS

Given a predictor matrix **X** (*N* x *V*), a response matrix **Y** (*N* x *M*) and the observation related to an incoming test sample $\mathbf{x}_{\text{new}}$ (1 x *V*):

1. Calculate the values of the distance between $\mathbf{x}_{\text{new}}$ and all samples in the calibration set **X**
2. Select the *K* nearest neighbours of $\mathbf{x}_{\text{new}}$ and construct the calibration subsets $\mathbf{X}_{\text{sub}}$ and $\mathbf{Y}_{\text{sub}}$
3. Assign the weight $w_{\text{local},n}$ to each *n*-th row of the matrix $\mathbf{X}_{\text{sub}}$

$$w_{\text{local},n} = exp\left(-\frac{d_n^*}{h\,\sigma(\mathbf{d}^*)}\right)$$

4. Calculate the diagonal matrix **D** by scaling and placing the elements of the vector $\mathbf{w}_{\text{local}}$ ($w_{\text{local},n}$) along its diagonal

$$\mathbf{D} = diag(\mathbf{w}_{\text{local}}) * \frac{1}{K}$$

5. Perform weighted mean-centering on $\mathbf{X}_{\text{sub}}$ and $\mathbf{Y}_{\text{sub}}$, with **1** being a vector of ones of appropriate size

$$\mathbf{X}_{\text{sub}} = \mathbf{X}_{\text{sub}} - \mathbf{1}\mathbf{1}^T\mathbf{D}\,\mathbf{X}_{\text{sub}}$$

$$\mathbf{Y}_{\text{sub}} = \mathbf{Y}_{\text{sub}} - \mathbf{1}\mathbf{1}^T\mathbf{D}\,\mathbf{Y}_{\text{sub}}$$

6. Initialize $\mathbf{u}_a$ as a column of $\mathbf{Y}_{\text{sub}}$
7. Calculate weighted **X**-weights

$$\mathbf{w}_a = \mathbf{X}_{\text{sub}}^T\mathbf{D}\,\mathbf{u}_a \left(\,\|\,\mathbf{X}_{\text{sub}}^T\mathbf{D}\,\mathbf{u}_a\,\|\,\right)^{-1}$$

8. Calculate **X**-scores

$$\mathbf{t}_a = \mathbf{X}_{\text{sub}}\,\mathbf{w}_a$$

9. Calculate **Y**-loadings

$$\mathbf{q}_a = \mathbf{Y}_{sub}^T\,\mathbf{D}\,\mathbf{t}_a \left(\mathbf{t}_a^T\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$$

10. Update $\mathbf{u}_a$

$$\mathbf{u}_a = \mathbf{Y}_{\text{sub}}\,\mathbf{q}_a$$

11. Repeat steps 7 to 10 until convergence of $\mathbf{u}_a$
12. Calculate **X**-loadings

(*continued on next page*)

(*continued*)

$\mathbf{p}_a = \mathbf{X}_{sub}^{T}\mathbf{D}\,\mathbf{t}_a\left(\mathbf{t}_a^{T}\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$

13. Deflate $\mathbf{X}_{sub}$ and $\mathbf{Y}_{sub}$

$\mathbf{X}_{sub} = \mathbf{X}_{sub} - \mathbf{t}_a\,\mathbf{p}_a^{T}$

$\mathbf{Y}_{sub} = \mathbf{Y}_{sub} - \mathbf{t}_a\,\mathbf{q}_a^{T}$

14. Repeat steps 7 to 13 until all the required latent variables (A) are extracted
15. Calculate regression coefficients

$\mathbf{B} = \mathbf{W}\left(\mathbf{P}^{T}\mathbf{W}\right)^{-1}\mathbf{Q}^{T}$

The prediction of the response associated with a new measured observation $\mathbf{x}_{new}$ is then obtained as:

$\hat{\mathbf{y}} = (\mathbf{x}_{new} - \overline{\mathbf{x}})\,\mathbf{B} + \overline{\mathbf{y}}$

with $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ being the vectors containing the column means of **X** and **Y**, respectively, estimated on the subset as described in Step 5

16. Repeat steps 1 to 15 for each new test sample

2.2.3. RoBoost-PLS

RoBoost-PLS [27,28] is a variant of classical PLS proposed by Metz et al. with the aim of reducing the impact of outliers during the model calibration phase.

RoBoost-PLS is initialised with weights identical for all the N investigated samples and equal to $1/N$. Once the first RoBoost-PLS latent variable or component has been extracted, each one of these weights is adjusted as:

$$w_{RoBoost,n} = \frac{1}{N}\; g(\|\boldsymbol{e}_n\|,\alpha)\;\cdot\;\prod_{j=1}^{m} g\left(f_{n,j},\beta\right)\cdot g(l_n,\gamma) \tag{5}$$

with $\|\ \|$ denoting the Euclidean norm, $\boldsymbol{e}_n$ being the **X**-residual vector associated to the n-th calibration sample, $f_{n,j}$ the j-th element of the **Y**-residual vector associated to the n-th calibration sample, l_n the leverage computed for the n-th calibration sample. Residuals and leverage are computed at step 11 of Algorithm 3. α, β, and γ in Equation (5) are adjustable hyperparameters that regulate the influence of the **X**-residuals, **Y**-residuals, and leverage, respectively, in the estimation of the RoBoost-PLS weights. The bisquare function g for a generic variable z_n is instead defined as:

$$g(z_n) = \begin{cases} \left(1 - z_n^2\right)^2 & for\ |z_n| < 1 \\ 0 & for\ |z_n| > 1 \end{cases} \tag{6}$$

For each metric, the argument z_n is calculated by normalising the quantity of interest by its median multiplied by the corresponding hyperparameter. Specifically:

$$\text{For } g(\|\boldsymbol{e}_n\|,\alpha),\ z_n = \frac{\|\boldsymbol{e}_n\|}{\alpha\cdot median(\|\boldsymbol{e}\|)} \tag{7a}$$

$$\text{For } g\left(f_{n,j},\beta\right),\quad z_n = \frac{f_{n,j}}{\beta\cdot median\left(\boldsymbol{f}_j\right)} \tag{7b}$$

$$\text{For } g(l_n,\gamma),\quad z_n = \frac{l_n}{\gamma\cdot median(\|\boldsymbol{l}\|)} \tag{7c}$$

g is one of the weighting functions most commonly used in robust calibration methods [23] and defines a weighting scheme that is based on the assumption that outliers tend to have extreme values in terms of residuals and/or leverage. By normalising these quantities with respect to their median, observations that deviate significantly from it will show normalized values greater than 1. As a result, they will be assigned a zero weight and, thus, effectively down-weighted (Equation (6)).

It is important to notice here that i) such an operation of weight readjustment is conducted every time a new RoBoost-PLS latent variable is derived, ii) the readjusted weights obtained for a given RoBoost-PLS component are used as weight initial estimates for the successive one and iii) α, β, γ are parameters of the function g that need to be tuned.

Algorithm 3: RoBoost-PLS

Given a predictor matrix **X** (N x V) and a response matrix **Y** (N x M)

1. Initialize $\mathbf{u_a}$ as a column of **Y**
2. Assign an equal initial weight $w_{RoBoost,i}$ to each n-th row of the matrix **X** as:

$$w_{RoBoost,n} = \frac{1}{N}$$

3. Calculate the diagonal matrix **D** by placing the elements of the vector $\mathbf{w}_{RoBoost}$ $(w_{RoBoost,n})$ along its diagonal
4. If $a = 1$, perform weighted mean-centering on **X** and **Y**, with **1** being a vector of ones of appropriate size

$\mathbf{X} = \mathbf{X} - \mathbf{1}\mathbf{1}^{T}D\ \ X$

$\mathbf{Y} = \mathbf{Y} - \mathbf{1}\mathbf{1}^{T}\mathbf{D}\,\mathbf{Y}$

5. Calculate weighted **X**-weights

$\mathbf{w_a} = \mathbf{X}^{T}\mathbf{D}\,\mathbf{u}_a\left(\ \|\ \mathbf{X}^{T}\mathbf{D}\,\mathbf{u}_a\ \|\ \right)^{-1}$

6. Calculate **X**-scores

$\mathbf{t}_a = \mathbf{X}\,\mathbf{w}_a$

7. Calculate **Y**-loadings

$\mathbf{q}_a = \mathbf{Y}^{T}\,\mathbf{D}\,\mathbf{t}_a\left(\mathbf{t}_a^{T}\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$

8. Update $\mathbf{u}_a$

$\mathbf{u}_a = \mathbf{Y}\,\mathbf{q}_a$

9. Calculate a convergence parameter proposed by *Metz* et al. in [28]

$\phi_a = \mathbf{u}_a\,\mathbf{D}\,\mathbf{t}_a\left(\mathbf{t}_a^{T}\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$

10. Calculate **X**-loadings

$\mathbf{p}_a = \mathbf{X}^{T}\mathbf{D}\,\mathbf{t}_a\left(\mathbf{t}_a^{T}\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$

11. Calculate **X**-residuals (**E**), **Y**-residuals (**F**) and leverage values (**l**)

$\mathbf{E} = \mathbf{X}^{T} - \mathbf{t}_a\,\mathbf{p}_a^{T}$

$\mathbf{F} = \mathbf{Y} - \mathbf{t}_a\,\mathbf{q}_a^{T}$

$\mathbf{l} = \mathbf{t}_a$

12. Update the weight of each i-th calibration sample

$$w_{RoBoost,n} = \frac{1}{N}\; g(\|\boldsymbol{e}_n\|,\alpha)\;\cdot\;\prod_{j=1}^{m} g\left(f_{n,j},\beta\right)\cdot g(l_n,\gamma)$$

13. Update **D** accordingly

$\mathbf{D} = diag(\mathbf{w}_{RoBoost})$

14. Repeat steps 4 to 13 until convergence of ϕ_a
15. Deflate **X** and **Y**

$\mathbf{X} = \mathbf{X} - \mathbf{t}_a\,\mathbf{p}_a^{T}$

$\mathbf{Y} = \mathbf{Y} - \mathbf{t}_a\,\mathbf{q}_a^{T}$

16. Repeat steps 5 to 15 until all the required latent variables (A) are extracted
17. Calculate regression coefficients

(*continued on next page*)

(*continued*)

$$\mathbf{B} = \mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{Q}^{\mathrm{T}}$$

The prediction of the response associated with a new measured observation $\mathbf{x}_{new}$ is then obtained as:

$$\hat{\mathbf{y}} = (\mathbf{x}_{new} - \overline{\mathbf{x}})\,\mathbf{B} + \overline{\mathbf{y}}$$

with $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ being the vectors containing the column means of $\mathbf{X}$ and $\mathbf{Y}$, respectively, estimated on the subset as described in Step 4.

2.2.4. Locally-weighted-RoBoost-PLS (LW-RoBoost-PLS)

2.2.4.1. Algorithmic scheme. LW-RoBoost-PLS relies on a synergistic combination of KNN-LW-PLS and RoBoost-PLS. This approach is mainly based on the algorithmic pipeline in Algorithm 2, but once defined the local calibration subset , $\mathbf{X}_{\mathrm{sub}}$, a RoBoost-PLS model (see Algorithm 3) is constructed instead of a standard PLS one. More specifically, LW-RoBoost-PLS encompasses the following 4 computational steps:

1) For each new sample whose responses are to be predicted, say $\mathbf{x}_{\mathrm{new}}$, its distance from any calibration sample is calculated. This distance can be calculated in the original variable space, in the subspace of a certain number of principal components of $\mathbf{X}$ or directly within the subspace of a global PLS model trained on $\mathbf{X}$ and $\mathbf{Y}$.
2) The K closest neighbours of $\mathbf{x}_{\mathrm{new}}$ are then identified and gathered in a new data matrix $\mathbf{X}_{\mathrm{sub}}$. Their respective responses are also collected in a new data array $\mathbf{Y}_{\mathrm{sub}}$;
3) The samples in $\mathbf{X}_{\mathrm{sub}}$ are weighted according to their distance to $\mathbf{x}_{\mathrm{new}}$.
4) RoBoost-PLS is applied to $\mathbf{X}_{\mathrm{sub}}$ and $\mathbf{Y}_{\mathrm{sub}}$. In this way, the local weights imposed in the previous step can be adjusted – but not increased to preserve the distance contribution – according to the degree of "outlyingness" of the corresponding observations. This step is crucial to mitigate the impact of possible abnormal samples present among the selected neighbours.

Algorithm 4: LW-RoBoost-PLS

Given a predictor matrix $\mathbf{X}$ (N x V), a response matrix $\mathbf{Y}$ (N x M) and the observation related to an incoming test sample $\mathbf{x}_{\mathrm{new}}$ (1 x V):

1. Calculate the values of the distance between $\mathbf{x}_{new}$ and all samples in the calibration set $\mathbf{X}$
2. Select the K nearest neighbours of $\mathbf{x}_{new}$ and construct the calibration subsets $\mathbf{X}_{\mathrm{sub}}$ and $\mathbf{Y}_{\mathrm{sub}}$
3. Assign the weight $w_{\mathrm{local},n}$ to each n-th row of the matrix $\mathbf{X}_{\mathrm{sub}}$

$$w_{\mathrm{local},n} = exp\left(-\frac{d_n^*}{h\,\sigma(\mathbf{d}^*)}\right)$$

4. Calculate the diagonal matrix $\mathbf{D}$ by scaling and placing the elements of the vector $\mathbf{w}_{\mathrm{local}}$ ($w_{\mathrm{local},n}$) along its diagonal

$$\mathbf{D} = diag(\mathbf{w}_{\mathrm{local}}) * \frac{1}{K}$$

5. If $a = 1$, perform weighted mean-centering on $\mathbf{X}_{\mathrm{sub}}$ and $\mathbf{Y}_{\mathrm{sub}}$

$$\mathbf{X}_{\mathrm{sub}} = \mathbf{X}_{\mathrm{sub}} - \mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{D}\,\mathbf{X}_{\mathrm{sub}}$$

$$\mathbf{Y}_{\mathrm{sub}} = \mathbf{Y}_{\mathrm{sub}} - \mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{D}\,\mathbf{Y}_{\mathrm{sub}}$$

6. Initialize $\mathbf{u}_a$ as a column of $\mathbf{Y}$
7. Calculate weighted $\mathbf{X}$-weights

$$\mathbf{w_a} = \mathbf{X}_{sub}^{\mathrm{T}}\mathbf{D}\,\mathbf{u}_a\left(\,\|\,\mathbf{X}_{sub}^{\mathrm{T}}\mathbf{D}\,\mathbf{u}_a\,\|\,\right)^{-1}$$

8. Calculate $\mathbf{X}$-scores

(*continued on next column*)

(*continued*)

$$\mathbf{t}_a = \mathbf{X}_{\mathrm{sub}}\,\mathbf{w}_a$$

9. Calculate $\mathbf{Y}$-loadings

$$\mathbf{q}_a = \mathbf{Y}_{sub}^{\mathrm{T}}\,\mathbf{D}\,\mathbf{t}_a\left(\mathbf{t}_a^{\mathrm{T}}\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$$

10. Update $\mathbf{u}_a$

$$\mathbf{u}_a = \mathbf{Y}_{\mathrm{sub}}\,\mathbf{q}_a$$

11. Calculate a convergence parameter proposed by *Metz* et al. in [28]

$$\phi_a = \mathbf{u}_a\,\mathbf{D}\,\mathbf{t}_a\left(\mathbf{t}_a^{\mathrm{T}}\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$$

12. Calculate $\mathbf{X}$-loadings

$$\mathbf{p}_a = \mathbf{X}_{\boldsymbol{sub}}^{T}\mathbf{D}\,\mathbf{t}_a\left(\mathbf{t}_a^{T}\,\mathbf{D}\,\mathbf{t}_a\right)^{-1}$$

13. Calculate $\mathbf{X}$-residuals ($\mathbf{E}$), $\mathbf{Y}$-residuals ($\mathbf{f}$) and leverage values ($\mathbf{l}$)

$$\mathbf{E} = \mathbf{X}_{\mathrm{sub}} - \mathbf{t}_a\,\mathbf{p}_a^{\mathrm{T}}$$

$$\mathbf{F} = \mathbf{Y}_{\mathrm{sub}} - \mathbf{t}_a\,\mathbf{q}_a^{\mathrm{T}}$$

$$\mathbf{l} = \mathbf{t}_a$$

14. Update the weights according to the degree of "outlyingness" of each i-th calibration sample

$$w_{\mathrm{RoBoost},n} = \frac{1}{N}\,g(\|\boldsymbol{e}_n\|,\alpha)\;\cdot\;\prod_{j=1}^{m} g\left(f_{nj},\beta\right)\cdot g(l_n,\gamma)$$

$$w_{\mathrm{RoBoost},n} = w_{\mathrm{local},n} \quad if \quad w_{\mathrm{RoBoost},n} > w_{\mathrm{local},n}$$

15. Update $\mathbf{D}$ accordingly

$$\mathbf{D} = diag(\mathbf{w}_{\mathrm{RoBoost}})$$

16. Repeat steps 5 to 16 until convergence of ϕ_a
17. Deflate $\mathbf{X}_{\mathrm{sub}}$ and $\mathbf{Y}_{\mathrm{sub}}$

$$\mathbf{X}_{\mathrm{sub}} = \mathbf{X}_{\mathrm{sub}} - \mathbf{t}_a\,\mathbf{p}_a^{\mathrm{T}}$$

$$\mathbf{Y}_{\mathrm{sub}} = \mathbf{Y}_{\mathrm{sub}} - \mathbf{t}_a\,\mathbf{q}_a^{\mathrm{T}}$$

18. Repeat steps 6 to 17 until all the required latent variables (A) are extracted
19. Calculate regression coefficients

$$\mathbf{B} = \mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{Q}^{\mathrm{T}}$$

The prediction of the response associated with a new measured observation $\mathbf{x}_{new}$ is then obtained as:

$$\hat{\mathbf{y}} = (\mathbf{x}_{new} - \overline{\mathbf{x}})\,\mathbf{B} + \overline{\mathbf{y}}$$

with $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ being the vectors containing the column means of $\mathbf{X}$ and $\mathbf{Y}$, respectively, estimated on the subset as described in Step 5

20. Repeat steps 1 to 19 for each new test sample

2.2.4.2. Parameter tuning. Several parameters need to be fine-tuned to attain an optimal LW-RoBoost-PLS model: the number of latent variables to extract, the number of nearest neighbours, K, as well as the values of h, α, β and γ. For this purpose, in this study, a double cross-validation procedure [36,37], consisting of two nested loops, was implemented: for each LW-RoBoost-PLS component, the inner loop was exploited to find the values of α, β and γ minimising the root median square error related to the responses of interest and defined as:

$$\mathrm{RMedSECV} = \sqrt{median\left[\left(\boldsymbol{y}_{CV} - \boldsymbol{y}_{true}\right)^2\right]} \tag{15}$$

where $\boldsymbol{y}_{CV}$ is a vector containing the response values predicted in cross-validation and $\boldsymbol{y}_{true}$ carries the reference response values.

The use of RMedSECV mitigates, in fact, the impact of potential outliers present in the calibration set. On the other hand, the outer loop was utilized to set h and k given the optimised α, β and γ. A grid search was conducted within the inner and outer loop, testing all possible combinations of the different hyperparameter values. A Måge plot [38] was ultimately resorted to for determining the final complexity of the LW-RoBoost-PLS model under study, *i.e.*, the number of LW-RoBoost-PLS latent variables associated to the best combination of h, K, α, β and γ. The Måge plot displays the RMedSECV values obtained for all possible combinations of tunable parameters as a function of the model complexity, highlighting for each number of latent variables (LVs) the combination yielding the lowest RMedSECV. Examples of such a plot are provided in the Supplementary Figs. S1 and S2.

It has to be noticed that, in order to guarantee the statistical rigour of the developed cross-validation approach (whose schematic representation can be found in Supplementary Fig. S3), the data splitting scheme adopted here was designed so as to ensure the independence of the subsets on which the inner and outer loops were run, respectively.

2.2.5. Model performance

The performance of the models resulting from the different approaches compared was evaluated in terms of root mean square error in external validation (RMSEP). It is important to note that the evaluation assumes that there are no outliers in the test set.

$$\mathrm{RMSEP} = \sqrt{\frac{\sum_{n=1}^{N}\left(y_{n,predicted} - y_{n,true}\right)^2}{N}} \tag{16}$$

and bias:

$$\mathrm{bias} = \frac{\sum_{n=1}^{N} y_{n,predicted} - y_{n,true}}{N} \tag{17}$$

3. Results and discussion

In this section, we compare the performance of LW-RoBoost-PLS with that of RoBoost-PLS and KNN-LW-PLS, focusing also on the way in which the different approaches assign weights to the calibration samples.

3.1. Simulated data

The optimal KNN-LW-PLS and RoBoost-PLS parameter combinations were determined through a 10-split venetian blind cross-validation approach, while for LW-RoBoost-PLS the double cross-validation procedure described in Section 2.2.4.2 was run with 5 data splits for both the inner and the outer loop. α, β and γ were all varied in the interval [3–7] with increments of 2 units, which resulted in 27 possible combinations to test. On the other hand, the range [30−100] with increments of 10 units and the values 1 and 2 were investigated for the local parameters h and K (for a total number of 16 possible combinations). Table 1 summarises the outcomes obtained for the three models optimised as detailed before, while Fig. 2 displays their corresponding prediction plots related to the external validation (test) set.

Table 1
Parameter settings of the compared models. LVs denotes the number of latent variables.

Model	Local Parameters (k, h)	RoBoost parameters (α, β, γ)	LVs	RMSEP	Bias
RoBoost-PLS		5, 3, 3	1	0.081	−0.035
KNN-LW-PLS	100, 1		2	0.062	0.097
LW-RoBoost-PLS	70, 2	3, 7, 5	2	0.049	0.000

Fig. 3, instead, represents the weights assigned to all the training samples by the three different methods compared in the present article. Mind that for KNN-LW-PLS and LW-RoBoost-PLS, only the weights of the single local model built for the specific sample highlighted as a red triangle in Fig. 2 are given. As one can clearly see, setting α, β and γ at 5, 3 and 3 allows RoBoost-PLS to correctly downweigh all the anomalous observations which were included in the calibration set (see Fig. 3a). It is worth noticing here that lower values of these parameters would have led to a more severe outlier detection, increasing the number of data items recognised as anomalous. However, since RoBoost-PLS is based on a linear modelling strategy, its prediction plot, as expected, shows a clear nonlinear trend (see Fig. 2a).

Conversely, by selecting a calibration subset of 100 neighbours per test sample (in this case, the distance values were calculated in the subspace of the first two principal components of **X**, explaining approximately 98 % of the variance of the entire calibration data), KNN-LW-PLS easily accommodates non-linearities, which is reflected by the fact that it leads to a lower RMSEP than RoBoost-PLS. Moreover, the nonlinear trend originally observed in Fig. 2a is significantly reduced when KNN-LW-PLS is exploited (see Fig. 2b). Nonetheless, the weights estimated by KNN-LW-PLS do not readily consider the existence of outliers among the K-nearest neighbours identified for each test observation. This might be the root cause inducing the higher prediction bias affecting KNN-LW-PLS.

Finally, LW-RoBoost-PLS permits to achieve a zero bias being capable of accurately down-weighing the outliers in the local calibration subsets selected. In their optimal combination, α, β, and γ were here found to be equal to 3, 7 and 5, respectively. Overall, these higher values compared to those resulting from the application of RoBoost-PLS may originate from the smaller number of outlying samples with which LW-RoBoost-PLS has to deal with iteratively when each of the aforementioned calibration subsets is handled. In addition to its inherent robustness against local outliers, LW-RoBoost-PLS can directly cope with the nonlinearities simulated in this circumstance (see also Fig. 2c), yielding the lowest RMSEP value among the three methodologies under study.

Furthermore, having a look at the first LW-RoBoost-PLS latent variable weights for a specific test sample (red triangle in Fig. 2c), it can be observed how this technique reduces the influence of samples exhibiting outlying behaviours even if they were chosen as members of the local calibration subsets. Step 14 of Algorithm 4 enables this by reducing or even nullifying the initial local weights, as indicated by the yellow stars in Fig. 3c. These weights are assigned based solely on the distance between the training observations and the test sample (blue stars in Fig. 3c). This adjustment occurs when the training observations exhibit abnormal **X**-residuals, **y**-residuals, and/or leverage. As represented, for instance, by the red circles in Fig. 3c). For illustration, the spectral profiles and simulated **y**-values for these four samples are displayed in Fig. 4, along with those of the remaining calibration subset considered in this specific scenario.

3.2. ABS data

The ABS data have recently been analysed by means of multiblock and local predictive approaches in an attempt to determine the influence of each production process stage on the quality of the manufactured copolymer. The original work conducted on them [30] encompassed several steps of data cleaning, outlier removal and response linearisation, which here were completely skipped in order to test the three methods under study in a rather complex industrial scenario. The raw data collected in the plant and fused as outlined in Section 2.1.2 were,

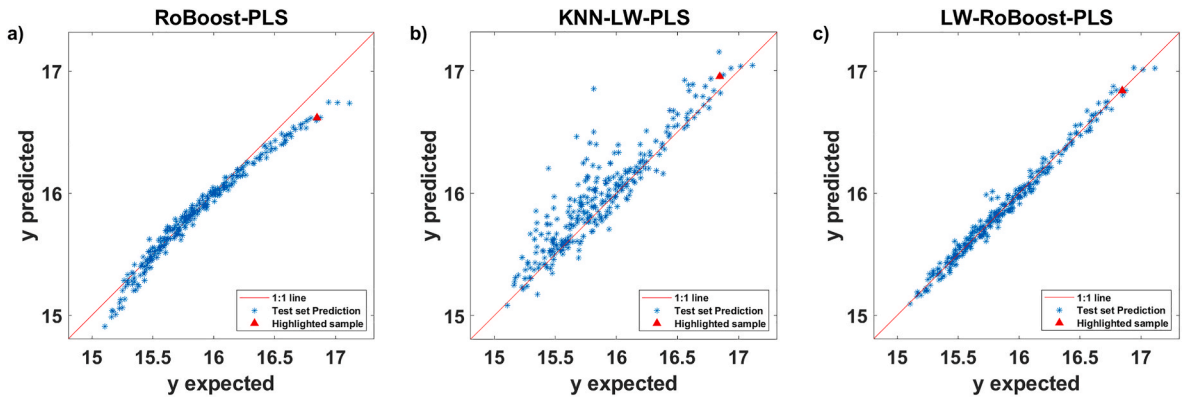


Fig. 2. Simulated data: predicted **y**-values versus measured **y**-values plots resulting from the application of the optimal **a)** RoBoost-PLS, **b)** KNN-LW-PLS and **c)** LW-RoBoost-PLS models. The displayed predictions relate to the samples of the external validation (test) set. For the sake of comparison, the predicted **y**-values obtained using a classical PLS model are reported in Supplementary Fig. S4.

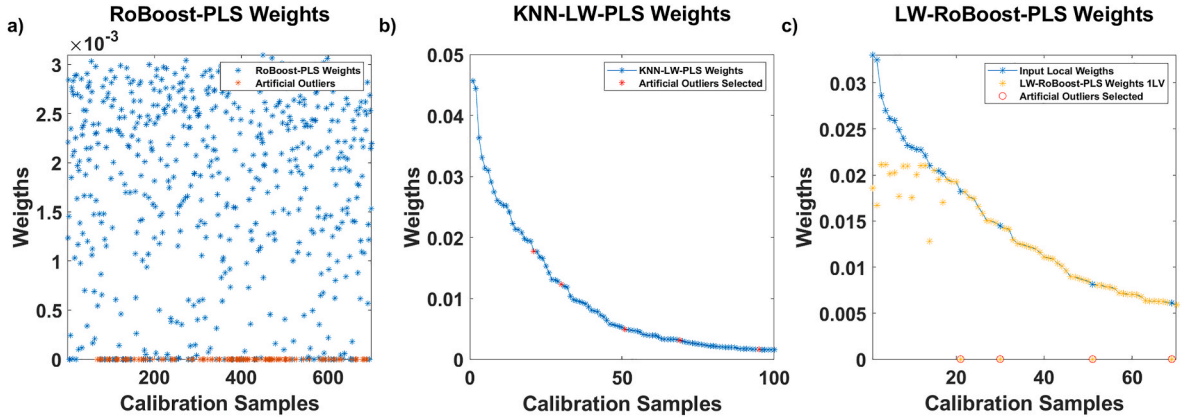


Fig. 3. Simulated data: calibration sample weights estimated by **a)** RoBoost-PLS, **b)** KNN-LW-PLS and **c)** LW-RoBoost-PLS. Notice that for KNN-LW-PLS and LW-RoBoost-PLS only the weights of the local models constructed for the sample denoted with a red triangle in Fig. 2b and c are given. In Fig. 3c, the blue stars represent the initial weights assigned based on the distance between the training observations and this test sample, while the yellow ones correspond to the final weights calculated at the end of the LW-RoBoost-PLS computational procedure. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

thus, directly input to RoBoost-PLS, KNN-LW-PLS and LW-RoBoost-PLS for comparison purposes. It is important to note that KNN-LW-PLS was replaced by *K*-Nearest-Neighbours-Locally-Weighted-Multiblock-PLS (KNN-LW-MB-PLS) [30], ensuring that the nearest neighbours identified for each test sample remained unchanged across all investigated data blocks. The same modification was also applied to LW-RoBoost-PLS, at least for steps 1 to 3 of Algorithm 4. Additionally, distinct preprocessing operations were performed on the data blocks: NIR spectra were pre-treated using Standard Normal Variate (SNV [39]) and mean centering, while measurements from each process sensor used throughout the production process were autoscaled. Prior to multiblock data analysis, the single blocks were finally scaled to unit block variance. In order to keep the calibration and cross-validation sets truly independent, pre-processing was performed after the cross-validation data splitting.

As for the simulated data, we report in Table 2 the outcomes related to the prediction of the response values associated to the external validation set samples returned by a RoBoost-PLS, a KNN-LW-MB-PLS and a LW-RoBoost-PLS model optimised as outlined in Section 3.1.

RoBoost-PLS extracts in total 4 LVs, but the retrieved combination of α, β and γ values (all set equal to 5) in this case seems not to effectively decrease the influence of all the outliers present in the calibration data as several predicted **y**-values it yields are far away from the ideal 1:1 fitting line (see Fig. 5a). On the other hand, KNN-LW-MB-PLS requires 200 neighbours and 4 LVs to build local models that allow to significantly improve the prediction quality (see, for instance, the blue dots in Fig. 5b) and guarantee a more reasonable **y**-residuals distribution (results not shown). Last but not least, LW-RoBoost-PLS outperforms both RoBoost-PLS and KNN-LW-MB-PLS in terms of predictive ability (it, indeed, returns the lowest RMSEP), as also the prediction plot in Fig. 5c highlights. At a closer look, it is also possible to notice how LW-RoBoost-PLS is capable of reducing the prediction bias affecting specifically the samples of product (grade) 5 and 8 as well as the dispersion observed when KNN-LW-MB-PLS was exploited, for example, for product 3 at high **y**-values.

In addition, Fig. 6 displays the weights assigned to all the training samples by the three considered approaches (once again, for KNN–KNN-LW-PLS and LW-RoBoost-PLS only the weights of a single local model – the one constructed for the test sample denoted with a red triangle in Fig. 5b and c – are given). In this circumstance, LW-RoBoost-PLS manages to zero some of the weights mistakenly kept higher by both RoBoost-PLS and KNN-LW-MB-PLS (see the red dots in Fig. 6a and b) and evidently associated to observations characterised by aberrant NIR spectral profiles and/or **y**-values (see Fig. 7). Notably, despite their local nature, models like those provided by LW-RoBoost-PLS preserve a direct link with the original measured variables, which can be leveraged for interpretability, as shown in Ref. [30]. An example of how to possibly interpret a LW-RoBoost-PLS model is illustrated in Supplementary Fig. S6.

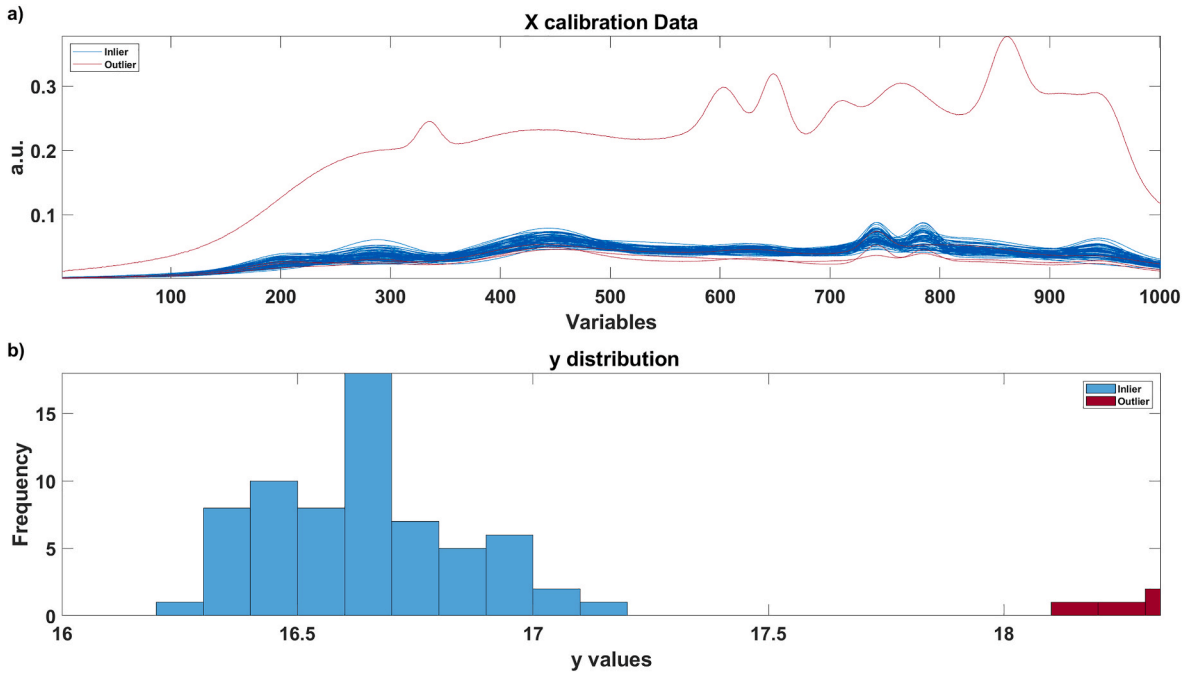


Fig. 4. Simulated data: **a)** pseudo-spectral profiles and **b)** **y**-values of the samples belonging the local calibration subset identified for the test observation denoted with a red triangle in Fig. 2b and c. Aberrant behaviours can be observed for the samples to which LW-RoBoost-PLS finally assigns a zero weight (see red solid lines and bars). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 2
Parameter settings of the compared multiblock (MB) models (for the sake of simplicity, MB has been omitted from the algorithms' acronyms in the first column). LVs denotes the number of latent variables.

Model	Local Parameters (*k*, *h*)	RoBoost parameters (α, β, γ)	LVs	RMSEP (g)	Bias (g)
RoBoost-PLS		5, 5, 5	4	0.85	−0.39
KNN-LW-PLS	200, 2		4	0.82	0.16
LW-RoBoost-PLS	200, 1	7, 3, 7	5	0.64	−0.05

4. Conclusions

In this article, we proposed a novel PLS-based multivariate calibration approach capable of handling non-linearities between predictors and responses while mitigating the detrimental influence of outliers during the calibration of a regression model. This approach, named LW-RoBoost-PLS, relies on a double weighting scheme resulting from a rational combination of the operational principles of KNN-LW-PLS and RoBoost-PLS. This combination is the key feature that makes it possible to simultaneously overcome the challenges posed by non-linearities and outliers.

LW-RoBoost-PLS was here tested on both simulated and complex real-world data and was found to outperform both the native methods from which it originates in terms of predictive power when the data under study exhibited strong non-linear variable intercorrelations and contained aberrant observations in **X** and/or **Y**. More specifically, LW-RoBoost-PLS proved to be particularly suitable for real-time prediction in industrial scenarios where the collected measurements are usually

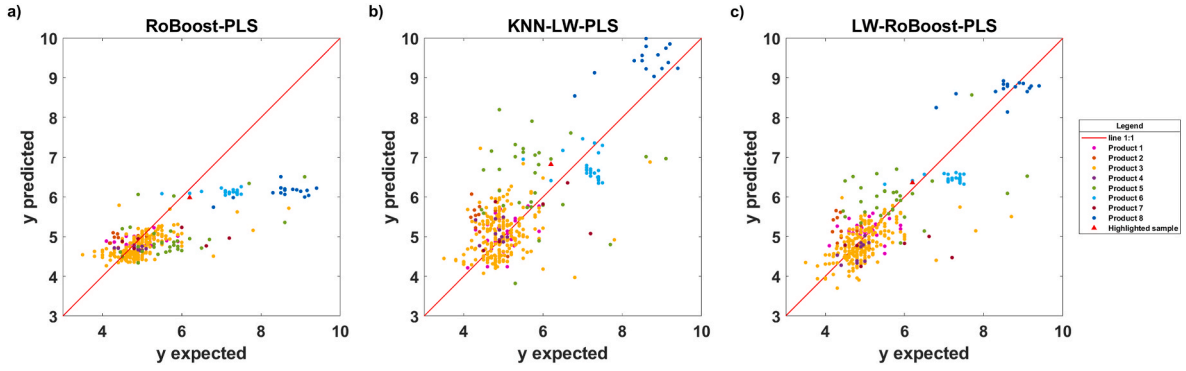


Fig. 5. ABS data: predicted **y**-values versus measured **y**-values plots resulting from the application of the optimal **a)** RoBoost-PLS, **b)** KNN-LW-MB-PLS and **c)** LW-RoBoost-PLS models. The displayed predictions relate to the samples of the external validation (test) set. The colour coding reflects the manufactured ABS grade. For the sake of comparison, the predicted **y**-values obtained using a classical PLS model are reported in Supplementary Fig. S5. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

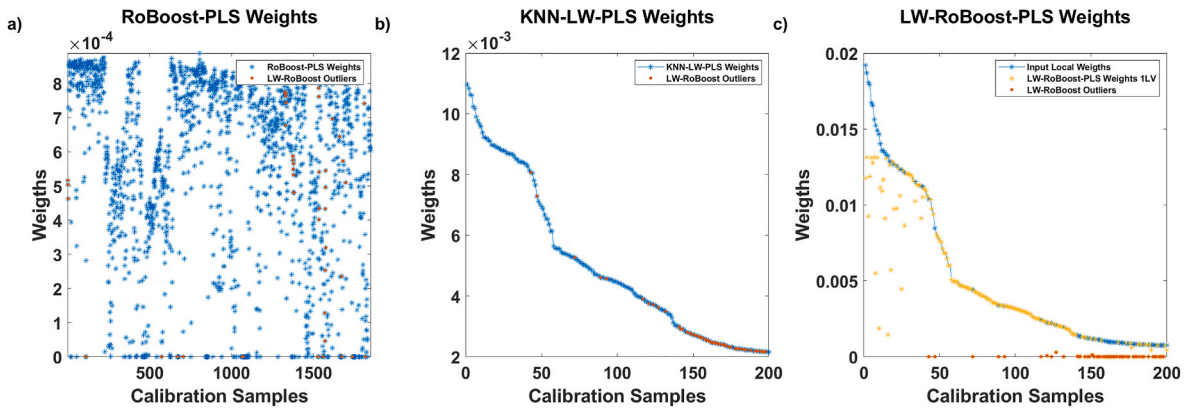


Fig. 6. ABS data: calibration sample weights estimated by **a)** RoBoost-PLS, **b)** KNN-LW-MB-PLS and **c)** LW-RoBoost-PLS. Notice that for KNN–KNN-LW-PLS and LW-RoBoost-PLS only the weights of the local models constructed for the sample denoted with a red triangle in Fig. 5b and c are given. In Fig. 5c, the blue stars represent the initial weights assigned based on the distance between the training observations and this test sample, while the yellow ones correspond to the final weights calculated at the end of the LW-RoBoost-PLS computational procedure. In Fig. 5a and b, the weights of the samples identified as outliers by LW-RoBoost-PLS (i.e., for which the LW-RoBoost-PLS weight was found to be approximately zero) are represented as red dots. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

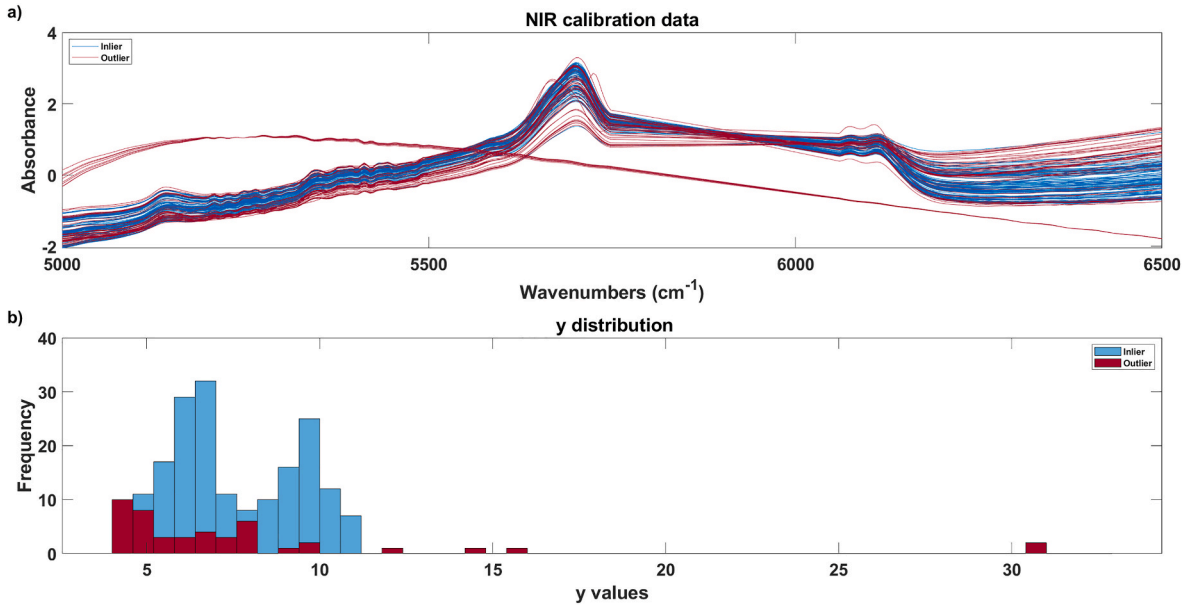


Fig. 7. ABS data: **a)** spectral profiles and **b)** **y**-values of the samples belonging to the local calibration subset identified for the test observation denoted with a red triangle in Fig. 5b and c. Aberrant behaviours can be observed for most samples to which LW-RoBoost-PLS finally assigns a zero weight (see red solid lines and bars). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

characterised by complex non-linearities related to seasonal variations or production scale-ups and by the presence of severe outliers due to sensor failures and/or drifts. It goes without saying then that fields such as pharmaceuticals – where spectroscopic techniques are widely used for drug formulation and process validation – or agriculture, where sensor data integration is critical for crop monitoring and soil analysis – may dramatically benefit from the robustness against non-linearities and outliers LW-RoBoost-PLS guarantees. Similarly, in environmental monitoring, where data streams from multiple sources often exhibit fluctuations and anomalies, LW-RoBoost-PLS could enhance predictive modelling and anomaly detection.

Concerning hyperparameter optimization, in this study the tested values were chosen based on prior knowledge and empirical experience with similar models. In general, for α, β and γ, higher values are typically used when a low number of outliers is expected and *vice versa*. If no prior information on the amount of outlying observations is available, larger intervals should be searched to ensure robustness. Regarding local parameters, the number of neighbours k should typically range from a few tens to a few hundreds, depending on the size and structure of the dataset.

In terms of computational efficiency, the proposed cross-validation strategy for model adjustment was found to be more demanding than running a single-step grid search accounting for all the six hyper-parameters altogether, but was preferred to preserve the independence of the validation subsets resorted to for tuning the local and the robust parameters, respectively. Nonetheless, the time required to process a new sample remains comparable to that of standard methods, typically well below 1 s. Moreover, when predicting response values for multiple new samples simultaneously, parallel computing strategies could be employed to further reduce execution time.

It is worth noticing that the method proposed in this article cannot readily cope with outlying test observations. This is issue will be addressed in future work.

CRediT authorship contribution statement

Daniele Tanzilli: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Lorenzo Strani:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Maxime Metz:** Writing – review & editing, Software, Resources. **Jean Michel Roger:** Writing – review & editing, Resources, Data curation. **Matthieu Lesnoff:** Writing – review & editing, Software. **Cyril Ruckebusch:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization. **Marina Cocchi:** Writing – review & editing, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Raffaele Vitale:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

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

Acknowledgments

Emilia Romagna Region supported PhD grant of one of the author (D. Tanzilli) fund: PA 2023-20467/RER CUP E83C23002540002

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.aca.2025.344167.

Data availability

Data will be made available on request.

References

[1] S. Wold, M. Sjöström, L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemometr. Intell. Lab. Syst. 58 (2001) 109–130, https://doi.org/10.1016/S0169-7439(01)00155-1.
[2] 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.
[3] S.V. Archontoulis, F.E. Miguez, Nonlinear regression models and applications in agricultural research, Agron. J. 107 (2015) 786–798, https://doi.org/10.2134/agronj2012.0506.
[4] A.H. Aastveit, P. Marum, Near-infrared reflectance spectroscopy: different strategies for local calibrations in analyses of forage quality, Appl. Spectrosc. 47 (1993) 463–469, https://doi.org/10.1366/0003702934334912.
[5] K. Fujiwara, M. Kano, S. Hasebe, A. Takinami, Soft-sensor development using correlation-based just-in-time modeling, AIChE J. 55 (2009) 1754–1765, https://doi.org/10.1002/aic.11791.
[6] P. Kadlec, R. Grbić, B. Gabrys, Review of adaptation mechanisms for data-driven soft sensors, Comput. Chem. Eng. 35 (2011) 1–24, https://doi.org/10.1016/j.compchemeng.2010.07.034.
[7] H. Martens, T. Næs, Multivariate Calibration, John Wiley & Sons, 1992.
[8] A. Höskuldsson, Quadratic PLS regression, J. Chemom. 6 (1992) 307–334, https://doi.org/10.1002/cem.1180060603.
[9] R.D. Cook, L. Forzani, PLS regression algorithms in the presence of nonlinearity, Chemometr. Intell. Lab. Syst. 213 (2021) 104307, https://doi.org/10.1016/j.chemolab.2021.104307.
[10] R. Rosipal, D. Clancy, Kernel partial least squares for nonlinear regression and discrimination. https://ntrs.nasa.gov/citations/20030014609, 2002. (Accessed 10 January 2024).
[11] K. Bennett, M. Embrechts, An optimization perspective on kernel partial least squares regression, Adv. Learn. Theory Methods Models Appl. 190 (2003).
[12] X. Zhang, W. Yan, H. Shao, Nonlinear multivariate quality estimation and prediction based on kernel partial least squares, Ind. Eng. Chem. Res. 47 (2008) 1120–1131, https://doi.org/10.1021/ie070741+.
[13] V. Centner, D.L. Massart, Optimization in locally weighted regression, Anal. Chem. 70 (1998) 4206–4211, https://doi.org/10.1021/ac980208r.
[14] D. Perezmarin, A. Garridovaro, J. Guerrero, Non-linear regression methods in NIRS quantitative analysis, Talanta 72 (2007) 28–42, https://doi.org/10.1016/j.talanta.2006.10.036.
[15] T. Naes, T. Isaksson, B. Kowalski, Locally weighted regression and scatter correction for near-infrared reflectance data, ACS Publ (2002), https://doi.org/10.1021/ac00206a003.
[16] J.S. Shenk, M.O. Westerhaus, P. Berzaghi, Investigation of a LOCAL calibration procedure for near infrared instruments, J. Near Infrared Spectrosc. 5 (1997) 223–232.
[17] F. Allegrini, J.A. Fernández Pierna, W.D. Fragoso, A.C. Olivieri, V. Baeten, P. Dardenne, Regression models based on new local strategies for near infrared spectroscopic data, Anal. Chim. Acta 933 (2016) 50–58, https://doi.org/10.1016/j.aca.2016.07.006.
[18] S. Kim, M. Kano, H. Nakagawa, S. Hasebe, Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection, Int. J. Pharm. 421 (2011) 269–274, https://doi.org/10.1016/j.ijpharm.2011.10.007.
[19] Z.-S. Duma, J. Susiluoto, O. Lamminpää, T. Sihvonen, S.-P. Reinikainen, H. Haario, KF-PLS: optimizing kernel partial least-squares (K-PLS) with kernel flows, Chemometr. Intell. Lab. Syst. 254 (2024) 105238, https://doi.org/10.1016/j.chemolab.2024.105238.
[20] F. Allegrini, A.C. Olivieri, Two sides of the same coin: Kernel partial least-squares (KPLS) for linear and non-linear multivariate calibration. A tutorial, Talanta Open 7 (2023) 100235, https://doi.org/10.1016/j.talo.2023.100235.
[21] R. Vitale, O.E. de Noord, A. Ferrer, Pseudo-sample based contribution plots: innovative tools for fault diagnosis in kernel-based batch process monitoring, Chemometr. Intell. Lab. Syst. 149 (2015) 40–52, https://doi.org/10.1016/j.chemolab.2015.09.013.
[22] M. Lesnoff, M. Metz, J.-M. Roger, Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data, J. Chemom. 34 (2020) e3209, https://doi.org/10.1002/cem.3209.
[23] R.J. Pell, Multiple outlier detection for multivariate calibration using robust statistical techniques, Chemometr. Intell. Lab. Syst. 52 (2000) 87–104, https://doi.org/10.1016/S0169-7439(00)00082-4.
[24] P. Filzmoser, V. Todorov, Review of robust multivariate statistical methods in high dimension, Anal. Chim. Acta 705 (2011) 2–14, https://doi.org/10.1016/j.aca.2011.03.055.
[25] P. Filzmoser, K. Nordhausen, Robust linear regression for high-dimensional data: an overview, WIREs Comput. Stat. 13 (2021) e1524, https://doi.org/10.1002/wics.1524.
[26] D.J. Cummins, C.W. Andrews, Iteratively reweighted partial least squares: a performance analysis by monte carlo simulation, J. Chemom. 9 (1995) 489–507, https://doi.org/10.1002/cem.1180090607.
[27] M. Metz, F. Abdelghafour, J.-M. Roger, M. Lesnoff, A novel robust PLS regression method inspired from boosting principles: roboost-plsr, Anal. Chim. Acta 1179 (2021) 338823, https://doi.org/10.1016/j.aca.2021.338823.
[28] M. Metz, M. Ryckewaert, S. Mas-Garcia, R. Bendoula, P. Dardenne, M. Lesnoff, J.-M. Roger, RoBoost-PLS2-R: an extension of RoBoost-PLSR method for multi-response, Chemometr. Intell. Lab. Syst. 222 (2022) 104498, https://doi.org/10.1016/j.chemolab.2022.104498.
[29] M. Metz, A. Biancolillo, M. Lesnoff, J.-M. Roger, A note on spectral data simulation, Chemometr. Intell. Lab. Syst. 200 (2020) 103979.
[30] D. Tanzilli, L. Strani, F. Bonacini, A. Ferrando, M. Cocchi, C. Durante, Implementing multiblock techniques in a full-scale plant scenario: on-line prediction of quality parameters in a continuous process for different acrylonitrile butadiene styrene (ABS) products, Anal. Chim. Acta 1316 (2024) 342851, https://doi.org/10.1016/j.aca.2024.342851.
[31] L. Strani, R. Vitale, D. Tanzilli, F. Bonacini, A. Perolo, E. Mantovani, A. Ferrando, M. Cocchi, A multiblock approach to fuse process and near-infrared sensors for On-Line prediction of polymer properties, Sensors 22 (2022) 1436, https://doi.org/10.3390/s22041436.
[32] L. Strani, E. Mantovani, F. Bonacini, F. Marini, M. Cocchi, Fusing NIR and process sensors data for polymer production monitoring, Front. Chem. 9 (2021) 748723, https://doi.org/10.3389/fchem.2021.748723.
[33] A.M.C. Davies, H.V. Britcher, J.G. Franklin, S.M. Ring, A. Grant, W.F. McClure, The application of fourier-transformed near-infrared spectra to quantitative analysis by comparison of similarity indices (CARNAC), Mikrochim. Acta 94 (1988) 61–64, https://doi.org/10.1007/BF01205839.
[34] S. Schaal, C.G. Atkeson, S. Vijayakumar, Scalable techniques from nonparametric statistics for real time robot learning, Appl. Intell. 17 (2002) 49–60, https://doi.org/10.1023/A:1015727715131.
[35] BhupinderS. Dayal, J.F. MacGregor, Improved PLS algorithms, J. Chemom. 11 (1997) 73–85, https://doi.org/10.1002/(SICI)1099-128X(199701)11:1<73::AID-CEM435>3.0.CO;2.
[36] D. Baumann, K. Baumann, Reliable estimation of prediction errors for QSAR models under model uncertainty using double cross-validation, J. Cheminf. 6 (2014) 47, https://doi.org/10.1186/s13321-014-0047-1.
[37] P. Filzmoser, B. Liebmann, K. Varmuza, Repeated double cross validation, J. Chemom. 23 (2009) 160–171, https://doi.org/10.1002/cem.1225.
[38] I. Måge, E. Menichelli, T. Næs, Preference mapping by PO-PLS: separating common and unique information in several data blocks, Food Qual. Prefer. 24 (2012) 8–16, https://doi.org/10.1016/j.foodqual.2011.08.003.
[39] Å. Rinnan, F.V.D. Berg, S.B. Engelsen, Review of the most common pre-processing techniques for near-infrared spectra, TrAC, Trends Anal. Chem. 28 (2009) 1201–1222, https://doi.org/10.1016/j.trac.2009.07.007.