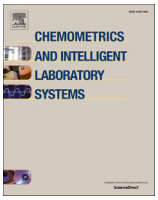
Chemometrics and Intelligent Laboratory Systems 243 (2023) 105002

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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



Comparison of machine learning approaches for the classification of elution profiles

Giacomo Baccolo [a,b], Huiwen Yu [b], Cecile Valsecchi [a], Davide Ballabio [a], Rasmus Bro [b,*]

[a] Department of Earth and Environmental Sciences, University of Milano-Bicocca, P.zza della Scienza, 1, 20126, Milano, Italy
[b] Department of Food Science, University of Copenhagen, Rolighedsvej 30, DK-1958, Frederiksberg C, Denmark

ARTICLE INFO

Keywords:
Chromatography
PARAFAC2
Neural networks
Automatic analysis

ABSTRACT

Hyphenated chromatography is among the most popular analytical techniques in omics related research. While great advancements have been achieved on the experimental side, the same is not true for the extraction of the relevant information from chromatographic data. Extensive signal preprocessing is required to remove the signal of the baseline, resolve the time shifts of peaks from sample to sample and to properly estimate the spectra and concentrations of co-eluting compounds.

Among several available strategies, curve resolution approaches, such as PARAFAC2, ease the deconvolution and the quantification of chemicals. However, not all resolved profiles are relevant. For example, some take into account the baseline, others the chemical compounds. Thus, it is necessary to distinguish the profiles describing relevant chemistry. With the aim to assist researchers in this selection phase, we have tried three different classification algorithms (convolutional and recurrent neural networks, k-nearest neighbours) for the automatic identification of GC-MS elution profiles resolved by PARAFAC2.

To this end, we have manually labelled more than 170,000 elution profiles in the following four classes: 'Peak', 'Cutoff peak',' Baseline' and 'Others' in order to train, validate and test the classification models.

The results highlight two main points: i) neural networks seem to be the best solution for this specific classification task confirmed by the overall quality of the classification, ii) the quality of the input data is crucial to maximize the modelling performances.

1. Introduction

Omics related research is rapidly increasing [1]. The omics approaches aim at a collective characterization of investigated samples. For instance, proteomics focuses on the analysis of the entire set of proteins for a given organism [2] and petroleomics studies the composition of petroleum at a molecular level [3].

This work is focused on metabolomics and related fields, such as foodomics and aroma analysis, where the objects of study are molecules with low molecular masses and hyphenated chromatography systems (e.g., GC-MS, LC-MS, LC-FTIR, GC-GC-MS LC-LC-MS) are gold standards for the quantification of such compounds [4].

On the analytical side, the omics experiments fall into two main categories, untargeted and targeted. The aim of the untargeted approach is the identification and relative quantification of as many compounds as possible within the analysed samples. Since the result of the experiment is an overview of a specific condition, it is interesting to compare different conditions, e.g., healthy/unhealthy, pure/adulterated, and this approach can be the starting point for hypothesis generation [5]. On the other hand, the targeted approach is focused on a predefined set of target molecules, up to tens or hundreds, and the aim is the absolute quantification of these compounds. The selection of these molecules is generally based on previous knowledge and the experiment is optimized for an accurate and reliable calibration in order to verify the experimental hypothesis [6]. The two approaches can be seen as complementary to each other: the hypothesis tested with a targeted analysis often comes from observations based on experiments performed with an untargeted approach.

Metabolomics experiments generate complex data both in terms of size, often in the order of gigabytes, and in terms of structure, since three or more dimensions are needed to store the results [4]. For example, when dealing with GC-MS measurements, a data matrix is obtained for each sample, where for each elution time the respective mass spectra are collected. Nowadays, the main bottleneck for omics studies is the

* Corresponding author.
E-mail address: rb@food.ku.dk (R. Bro).

https://doi.org/10.1016/j.chemolab.2023.105002
Received 14 July 2023; Received in revised form 27 September 2023; Accepted 28 September 2023
Available online 7 October 2023

analysis rather than the generation and acquisition of the data [7], resulting in the need for computational solutions able to speed up the analysis assisting the researchers.

As mentioned before, the complexity of the measured data is an issue especially when dealing with untargeted approaches. Untargeted data often suffer from drift of the baseline signal, coelution of different compounds, and retention time shifts of peaks from sample to sample [8,9]. However, it is difficult to optimize the experimental settings when the aim is to identify and quantify as many compounds as possible with acceptable accuracy. A common practice to simplify the analysis consists in defining intervals on the retention time dimension, focusing only on the relevant regions of the chromatogram, i.e., the peaks, in order to identify and quantify the corresponding compounds.

A number of different tools are available for the deconvolution and extraction of the relevant signals from GC-MS data [10]. A common deconvolution approach is Parallel Factor Analysis 2 (PARAFAC2) [11]. This modelling approach is based on the deconvolution of experimental signals [12,13] to automatically extract and separate the different contributions from the raw data, such as the elution profiles of the baseline and the peaks, as well as co-eluting peaks. The main advantages of this approach are the reproducibility of the results, which is user independent, and the effectiveness to extract the pure signals (elution profiles and mass spectra), increasing the accuracy of both the quantification and the identification of chemical compounds [14].

Despite the efficiency of PARAFAC2, the deconvoluted signals have to be manually checked, in order to assess if all the contributions have been separated or more PARAFAC2 components are needed, and also to identify the components that are describing the chemical compounds, i.e., the components that are describing the peaks. This step may be time-consuming, and it can introduce bias which depends on the user selection.

Expanding the results in Ref. [15], we carried out a comparison of different machine learning approaches for the classification of chromatographic profiles deconvoluted by means of PARAFAC2, with the aim to i) speed up the inspection of the resolved profiles, ii) avoid user bias and iii) propose an effective and automatic tool to assist researchers in the selection of the resolved profiles. These aspects are crucial to increase the reproducibility of the results. To this end, in this paper we produced a set of semi-quantitative criteria for assessing the quality of manually labelled data in order to increase the comparability of future developments in this field of research.

The classification approaches were trained and validated using a data set including more than 170,000 elution profiles resolved by means of PARAFAC2 and manually labelled accordingly in four classes: 'Peak', 'Cutoff peak',' Baseline' and 'Others'. We tested three different classification strategies: k-nearest neighbours (kNN) and two deep learning networks based on convolutional and recurrent neural networks, respectively. Moreover, results were compared with those obtained with a previously published convolutional neural network proposed for the same aim [15].

2. Materials and methods

2.1. Data

Head-space solid-phase microextraction (HS-SPME) GC-MS data obtained from 66 olive oil samples were used in this work. The experimental details are available elsewhere [16]. Briefly GC-MS data are organized in a three-way array corresponding to the elution time scans, the *m/z* s and the sample. In this case the dimensions of the datasets are 17,849 elution scans x 271 measured *m/z* x 66 samples.

2.2. PARAFAC2 theory

PARAFAC2 is a multiway factorization approach inspired by the concept of parallel proportional profiles introduced by Cattel [17]. Cattel affirmed that the different signals underlying a given system could be described by unambiguous components when the signals are constant across different samples but in different proportions. A PARAFAC2 model can be formalized as follow:

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{B}_k^{\mathrm{T}} + \mathbf{E}_k \qquad \text{Eq. 1}$$

where $\mathbf{X}_k$ is the *k*-th slab of the three-dimensional array $\underline{\mathbf{X}}$ with dimensions I × J × K. It derives that the matrix $\mathbf{X}_k$ has $I \times J$ dimensions containing the chromatographic run of sample *k* out of *K* samples. The x_{ij} element of $\mathbf{X}_k$ is the intensity measured at *m/z i,* at elution time point *j*, over all *I m/z* values and *J* time points.

A is an $I \times F$ matrix where the *f*-th column contains the resolved mass spectrum of factor *f* of each of the *F* components included in the PARAFAC2 model.
$\mathbf{D}_k$ is a $F \times F$ diagonal matrix holding the scores of sample *k* for each factor *f*.
$\mathbf{B}_k$ is a $J \times F$ matrix holding the elution profiles, where the *f*-th column contains the resolved elution profile of factor *f* for sample *k*. These resolved elution profiles have been then used as input (independent variables) to train the classification models. The cross-product of each $\mathbf{B}_k$ is required to be constant in PARAFAC2.

PARAFAC2 solutions are unique under mild conditions and details about the method and its properties can be found in Ref. [18].

2.3. Training, internal validation set and external test set

A total of 44 intervals on the time dimension have been defined and all the intervals were resolved by PARAFAC2 modelling. A total of 306 PARAFAC2 models were calculated, resulting in 1,214 components.

A total of 80,124 resolved elution profiles were obtained (1,214 components x 66 samples) and then used as input for the classification models. These profiles were randomly split into two sets: 68,106 profiles (85 %) were included in the training set and the remaining 12,018 profiles (15 %) in the validation set. The models were trained on the training set and their hyperparameters were tuned by minimizing the prediction error on the validation set. To evaluate the predictive ability of the trained models, we considered an external test set containing made of 7,673 profiles not from the olive oil dataset, preprocessed with PARAFAC2, as the training data. These profiles have been retrieved as a subset from the test set in Ref. [14].

To increase the number and the variability of the data, all the profiles have been duplicated by horizontally flipping them, resulting in a total of 136,212 training, 24,036 validation and 15,346 test profiles, respectively.

2.4. Labelling

All the profiles have been manually labelled according to four classes: 'Peak', 'Cutoff peak', 'Baseline' and 'Other'. First, each profile was linearly interpolated to a length of 50 points and normalized to unit vector, i.e., vectors with norm equal to one. This preprocessing equalizes the dimensionality of the elution profiles and, thus, allows the labelling of profiles with different lengths. At the same time, the normalization maximizes the comparability of the profiles. The preprocessing has been adopted according to the preprocessing routine defined in Ref. [14]. In order to be consistent during the manual evaluation of the elution profiles, we have defined a set of rules, represented in Fig. 1, to assist the visual labelling of the profiles.

The rules applied during the manual labelling were the following:

- A profile visually assessed as a peak, was retained as such if the main peak had a normalized maximum intensity higher than 0.1, while the

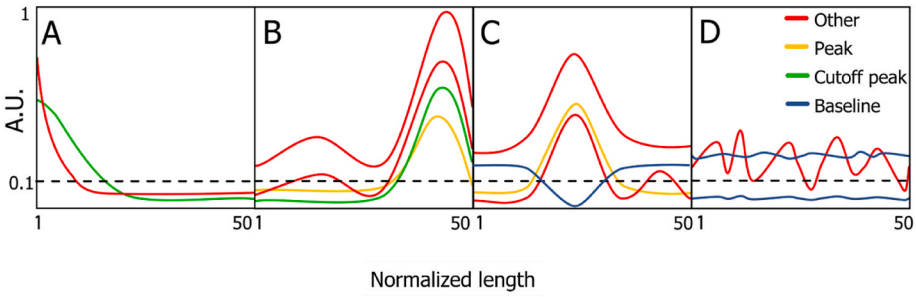


Fig. 1. Graphical representation of the labelling criteria. The dotted line highlights the 0.1 threshold.

rest of the profile had a normalized intensity lower than 0.1 (Fig. 1 B, C)

- A profile visually assigned to the class 'Cutoff peak' was retained as such when the main peak had a normalized maximum intensity higher than 0.1 (Fig. 1 A, B), and a clear inflection point was visible (Fig. 1 A, B); while the rest of the profile had a normalized intensity lower than 0.1.
- A profile was assigned to the class 'Baseline' when a flat or monotonically increasing or decreasing profile was present (considering reasonable noise), (Fig. 1 D), with a difference between the maximum and minimum signal value smaller than 0.2. Baseline profiles may show a negative peak (Fig. 1C).
- A profile was assigned to the class 'Other' if it did not meet the criteria for the above classes.

It should be noted that this set of rules does not include any indication about the overall shape for any of the considered classes: this means that the rules cannot be automatically applied for the labelling of the profiles.

All the profiles were thus manually labelled by visual inspection and with the application of the semi-quantitative rules. Details about the class distribution and the number of profiles for the training, validation and test set are shown in Table 1.

2.5. Classification models

A brief overview of the theory behind the classification models used in this study is given, together with details about their optimization.

2.6. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a family of neural networks widely applied in image analysis. Several different variations in CNN architectures have been proposed, but in general they consist of stacked convolutional and pooling layers, followed by one or more fully connected layer(s). The convolutional layer is the core of a CNN, and it is based on a set of trainable filters or kernels. Basically, it can be seen as a pattern extractor. The filter considers only a portion of the input data to find specific parts.

The inputs are convolved with the weights, which are optimized in the training phase, to obtain a new feature map. The result of this operation is the input for an activation function. Formally the convolution operation can be written as:

$$\mathbf{Y}_k = f(\mathbf{W_k} \times \mathbf{X}) \quad \text{(Eq. 2)}$$

where $\mathbf{X}$ is the input matrix or the output of the previous layer and $\mathbf{W_k}$ is the k-th filter related to the k-th feature map $\mathbf{Y_k}$ and f represents the activation function.

The pooling layer reduces the dimension of the feature map through information compression. There are two main strategies: max and average pooling. In the former case the pooling will extract the maximum value while in the latter the values are averaged.

The convolutional and pooling layers are followed by fully connected layer(s), which interprets the features selected by the previous layers. For classification problems the last layer uses a SoftMax operator which is a function providing a normalized probability distribution over the possible classes, four in our case.

In this paper the architecture of the CNN model has been retrieved by Ref. [15]. The network is made of four convolutive layers followed by two dense layers. Details about the settings and optimization of the CNN model can be found in Ref. [15].

Table 1
Class distribution and number of profiles included in the training, validation, and test sets.

Set	Other	Peak	Cutoff peak	Baseline	Total
Training	30.5 %	18.9 %	6.6 %	44 %	136,212
	41,528	25,794	8,932	59,958	
Validation	29.7 %	19.5 %	6.4 %	44.3 %	24,036
	7,150	4,698	1,548	10,640	
Test	34.5 %	24 %	15.1 %	26.4 %	15,346
	5,296	3,682	2,312	4,056	

2.7. (Bilinear)Long short-term memory

Recurrent neural networks (RNNs) are a family of neural networks used to deal with sequential data [19]. In this work, we used RNNs with Long short-term memory (LSTM) units, which were proposed to solve the vanishing and exploding gradient problem affecting the training of vanilla RNNs [20] and to capture long-range dependencies between sequence data. In speech recognition, where LSTM networks are widely used, long-range dependencies are important since the meaning of a sentence change depending on how the words are arranged, so keep track of the reciprocal positions of the words, even when they are not immediately next to each other, is crucial to understand the sense of the phrase [21]. The same concept can be applied in our context. For instance, let consider a 'Peak' and a 'Cutoff peak': in general, these profiles are both characterized by the same patterns, e.g., peak or flat curves. The difference between a peak and a cutoff peak is how these patterns are placed through the profile. Therefore, it is important to 'remember' how the different, and even distant, parts of a profile are organized. The LSTM networks are specifically designed to handle this kind of task.

RNNs compute a hidden vector $\mathbf{h}$, which is updated at each *t-th* time step as follows:

$$h_t = tanh(\mathbf{W}h_{t-1} + \mathbf{I}x_t) \quad \text{(Eq. 3)}$$

where tanh is the hyperbolic tangent function, $\mathbf{W}$ is the recurrent weight matrix, $\mathbf{I}$ is a projection matrix and x_t is the t-th element of the vector $\mathbf{x}$ ($1 \times T$) where T is equal to 50 (normalized length of the profiles). The hidden state $\mathbf{h}$ is used to compute $\mathbf{z}$, the output of the RNN cell:

$$z_t = softmax(\mathbf{W}h_{t-1}) \quad \text{(Eq. 4)}$$

By using **z** as the input to another RNN, different RNNs can be stacked together or with traditional fully connected layers creating deeper architectures and allowing to predict the class label for each profile.

When applied to profiles, $\mathbf{x} = \{x_1, x_2, \ldots x_T\}$, RNN layers process one x_t at a time, based on the preceding portions of the profile and a probability estimation. More specifically, RNNs model a dynamic system, where the hidden state (h_t) of the network at any t-th position in the profile is not only dependent on the current observation (x_t), but also relies on the previous hidden state (h_{t-1}).

The core of the LSTM is a memory cell, m_t, which represents the information of the inputs observed to the current time-step. The LSTM cell has the same inputs (i.e., the previous hidden state h_{t-1} and the input x_t) and provides the same outputs (h_t and z_t) as a vanilla RNN cell but controls the information flow by memory gates (update, forget and output gates). In particular, the forget and update gates determine the information to keep for later stages, by updating m_t. The output gate computes the outputs as functions of x_t, h_{t-1} and the memory cell vector m_{t-1}. This setting allows LSTMs to retain important features detected during earlier stages in the sequence, thereby capturing long-distance dependencies.

The most common version of LSTM is unidirectional, i.e., the input features are processed from left to right (forward direction). In this work, we also considered bidirectional LSTM (BILSTM) which allows simultaneous forward and backward prediction [22].

2.8. (BI)LSTM optimization

The general architecture for all the recurrent models tested during the optimization included the following layers: an input layer, (BI)LSTM layer(s), a fully connected layer and an output layer. This architecture implies the tuning of the hyperparameters listed in Table 2. In particular, two values of initial learning rate were considered (0.001 and 0.01). The input layer is made of 50 neurons (i.e., the length of the normalized profiles) and this was kept constant across all the tested networks. We have tried one, two or three stacked layers of (BI)LSTM followed by one fully connected layer. The same layer type has been used for each layer, therefore no combinations of LSTM and BILSTM layers have been tested in multiple layers models.

The number of neurons in the first layer was set to 16 or 32 or 48, and the number of neurons was halved each time for each successive (BI) LSTM layer, i.e., the number of neurons for the second and third layer was half and a fourth of the neurons in the first layer, respectively.

To avoid overfitting, we introduced a dropout of 0.25 as a regularization term for the first layer, but we also considered unregularized networks (i.e., dropout equal to 0).

The number of neurons in the fully connected layer is the same as the last (BI)LSTM layer. Three different activation functions have been considered: ReLU, sigmoid and hyperbolic tangent. The output layer with a SoftMax function (four neurons, one for each class) has been kept constant across all the models. Two optimizations algorithms or solvers have been tried: Adam and RMSProp [23].

A preliminary optimization of the hyperparameters has been performed by means of a full grid search considering the combination of all the parameters listed in Table 2, for a total of 432 tested networks. The ten architectures with the best classification performances (highest Non-Error Rate on the validation set, see next paragraph) have been selected; each network was replicated five times to test its stability. Since no significant differences have been found across the five replicas of the same architecture (Fig. S1), the overall best model considering the NER in validation has been selected. The final parameters for the selected model are shown in Table 2.

Table 2
Optimized and selected parameters for the RNN models.

Parameters	Option 1	Option 2	Option 3	Selected Option
Initial learning rate	0.001	0.01		0.001
Neurons Type	LSTM	BILSTM		BILSTM
Dropout	0	0.25		0.25
Solver	Adam	RMSprop		RMSprop
Activation function	ReLu	tanh	sigmoid	tanh
Num of (BI)LSTM Layers	1	2	3	3
Num of Neurons in the first (BI)LSTM layer	16	32	48	48

2.9. K nearest neighbours

The k Nearest Neighbours (kNN) algorithm is a benchmark classification method [24]. A sample is classified according to the most represented class among the k nearest training samples (neighbours). The Euclidean metric has been used for the distance calculation. The optimal number of neighbours k has been optimized testing different values from 1 to 10, 20, 30, 40 and 50: the optimal k value (4) was found minimizing the classification error on the validation set (Fig. S2).

2.10. Classification diagnostics

The classification performance has been evaluated by means of confusion matrices and derived measures [25]. The classification results can be summarized in the so-called confusion matrix, which is a $G \times G$ matrix, where G corresponds to the number of modelled classes. Each element c_{kg} of the confusion matrix represents the number of samples belonging to class k predicted as class g.

The sensitivity (Sn_g) for the gth class is defined as:

$$sn_g = \frac{c_{gg}}{n_g}, \quad \text{(Eq. 5)}$$

where c_{gg} is the number of samples of the gth class correctly classified and n_g corresponds to the total number of samples that belong to the gth class. The Non-Error Rate (NER, also known as Balanced Accuracy) is defined as the mean of class sensitivities:

$$NER = \frac{\sum_{g=1}^{G} Sn_g}{G} \quad \text{(Eq. 6)}$$

The precision (Pr_g) corresponds to the ratio of samples of class g correctly classified over the number of the samples predicted into the gth class:

$$Pr_g = \frac{c_{gg}}{n_g'} \quad \text{(Eq .7)}$$

As such the precision is used to quantify how many of the samples predicted as class g are actually belonging to that class.

Receiver Operating Characteristics (ROC) curves are a graphical tool for the diagnosis of a classification model [26]. The curve for a given class g is obtained by plotting the False Positive Rate (FPR) versus Sn_g, also known as True Positive Rate (FPR), as a function of a moving classification threshold. The AUC corresponds to the value of the area under the ROC curves.

2.11. Software

The PARAFAC2 models have been calculated with PARADISe [14] version 5.8, available at (https://ucphchemometrics.com/paradise/ (Sep 27, 2023). The models have been calculated with the non-negative fast algorithm [27].

All the classification measures have been calculated by means of routines in the classification toolbox for MATLAB [28], available at https://michem.unimib.it/download/matlab-toolboxes/classification-toolbox-for-matlab/(Nov 2, 2021).

The computations, optimization, training and test of the models have been performed in MATLAB (MATLAB 2021a, The MathWorks, Inc. Natick, Massachusetts, United States). The deep neural networks have been calculated with the MATLAB deep network designer toolbox.

3. Results and discussion

The classification performances obtained on the training, validation and the external test set are summarized in Fig. 2 in terms of NER. All the NER values are reported in Table S1. Overall, the classification performances of the models can be considered satisfactory.

NERs are always higher than 85 % for all the models when looking at training, validation and test sets. The kNN model has the highest NER considering both training and validation sets, which might be expected, considering the model structure, the low number of neighbours and how the two sets have been produced. However, the kNN model shows the biggest variation in NER values between the training set and the other sets, in particular the NER value decreases of 3 % and 7 % for the validation and the test set, respectively.

The deep learning models are characterized by more stable results when looking at the performances obtained on the training and validation sets. For the CNN model the NER for the training is slightly higher than that of the validation set (+0.5 %) and lower with respect to the test set (−2.6 %). Considering the BILSTM model, the difference between the NER values is 0.2 % between training and both validation and test sets.

3.1. Diagnostics of the results on the external test set

In order to further compare the classification performances of the three models on the external test set, the aggregated confusion plot, the NER, the class sensitivities and precisions of the three models achieved on the external test set are shown in Fig. 3. All the values for these classification measures are listed in Table S1.

The aggregated confusion plot derives from the combined analysis of the classification results by the three models. In this plot, the agreement across the three models is reported in a particular representation of a confusion matrix with a Venn-like diagram.

A given profile can be classified as belonging to a given class by i) all the three models (one possible combination), ii) two models (three possible combinations), or iii) only a single model (for a total of three models), giving a total of seven possible combinations.

Each profile was assigned to one or more of these combinations according to the concordance/discordance of the predictions with respect to the experimental class. In the $i \times j$ cell of the aggregated confusion plot, the number of profiles from the i-th class predicted as class j is represented in a graphical way, considering the seven combinations mentioned above. It means that, for each cell, seven numbers, one for each combination, should be given.

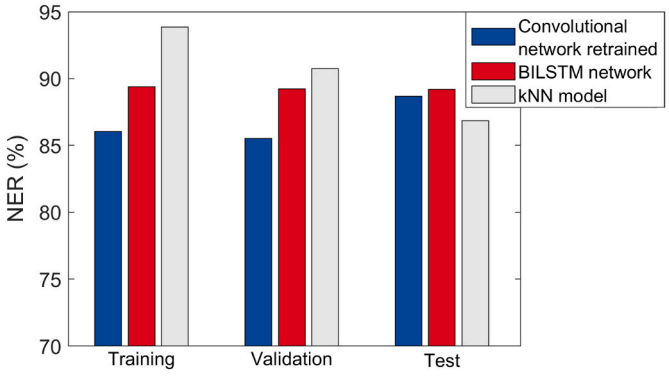


Fig. 2. NER values for the three trained models (CNN, BILSTM, kNN) considering the training, validation and test sets.

So, for instance, the 1 × 1 cell of the aggregated confusion plot reports how many profiles labelled as 'Other' were correctly predicted as 'Other' by: (1) all the models, i.e., all concordant predictions, (2) CNN and BILSTM models, (3) BILSTM and kNN models, (4) kNN and CNN models, or uniquely by (5) CNN or (6) BILSTM or (7) kNN. For ease of visualization, within each cell of the aggregated confusion plot we defined seven regions, one for each of the possible combinations. Specifically, for each square we defined: (1) a black central square that represents the concordant predictions across all the models, (2) a purple area for the CNN and BILSTM models concordant predictions, (3) a pink region for the BILSTM and kNN models concordant predictions, (4) a light blue area for the kNN and CNN models concordant predictions and (5) blue, (6) red and (7) grey area for CNN, BILSTM and kNN predictions, respectively.

The actual number of profiles for each area is reported on each areaFigure.

From the aggregated confusion plot represented in Fig. 3, it is possible to see that the black areas have the highest values on the diagonal. This suggests that most often the models classify the profiles consistently to each other and that the predictions are correct, as expected looking at the NER values reported in the right bottom square (Fig. 3 cell 5 × 5, Table S1) and in Fig. 2.

Considering the sensitivity for the class 'Other', the CNN and the BILSTM models perform better compared to the kNN model (cell 1 × 5, Fig. 3). This can be explained looking at the values in the grey and purple areas in cell 1 × 1 (Fig. 3) indicating that the CNN and the BILSTM correctly classify many more profiles belonging to this class compared to kNN. It means that the same profiles are assigned to a wrong class by the kNN model. The difference between CNN and BILSTM in terms of sensitivities is due to the set of profiles correctly classified uniquely by the CNN model (blue area in cell 1 × 1 Fig. 3).

Most of the misclassified 'Other' profiles by the kNN model are assigned to the class 'Peak' while for the BILSTM model the errors are equally distributed between the class 'Peak' and 'Baseline', as suggested by the higher values of the grey, pink and red areas in the cell 1 × 2 and 1 × 4 (Fig. 3). Taking advantage of the aggregated confusion plot, we have been able to easily identify these profiles. In most of the cases, the profiles resemble a peak but do not fulfil the criteria applied during the labelling. For instance, these profiles show some spikes beyond the 0.1 threshold in addition to the main peak. Another subset of these misclassified profiles is characterized by a certain amount of noise.

Looking at the precision for the class 'Other' obtained by the three models (cell 5 × 1 Fig. 3), it is possible to notice that the trend is the opposite compared to the sensitivities. In this case, the kNN achieved the best result, while the CNN the worst and still the BILSTM in the middle. The smaller precisions for the two deep learning methods are reflected in the aggregated confusion plot. The high value of the blue, pink and red areas in the 2 × 1 and 3 × 1 cells in Fig. 3 suggests that BILSTM and CNN tend to classify as 'Other' more profiles belonging to different classes compared to the kNN model. Visually inspecting these profiles, we found that in most of the case there is some residual noise, and the misclassification can be related to that. This observation supports the hypothesis that the deep leaning models are overestimating the influence of noise for the classification of the profiles while the KNN is less sensitive to this aspect.

Considering the class 'Other', the difference of the sensitivities between the CNN and the kNN model is 25%, while in terms of precision the difference between the kNN and CNN is around 4 %. While the lower precision of the deep learning models can be related to an overestimation of the noise influence, the lower sensitivity of the kNN model can indicate that kNN is underestimating the same aspect.

Moreover, considering the CNN and kNN models, the difference of the sensitivities and the precisions suggests that the underestimation of the kNN is more pronounced compared to the overestimation of the CNN. The BILSTM seems the more balanced model. This different behaviour of the models influences the results for the remaining classes,

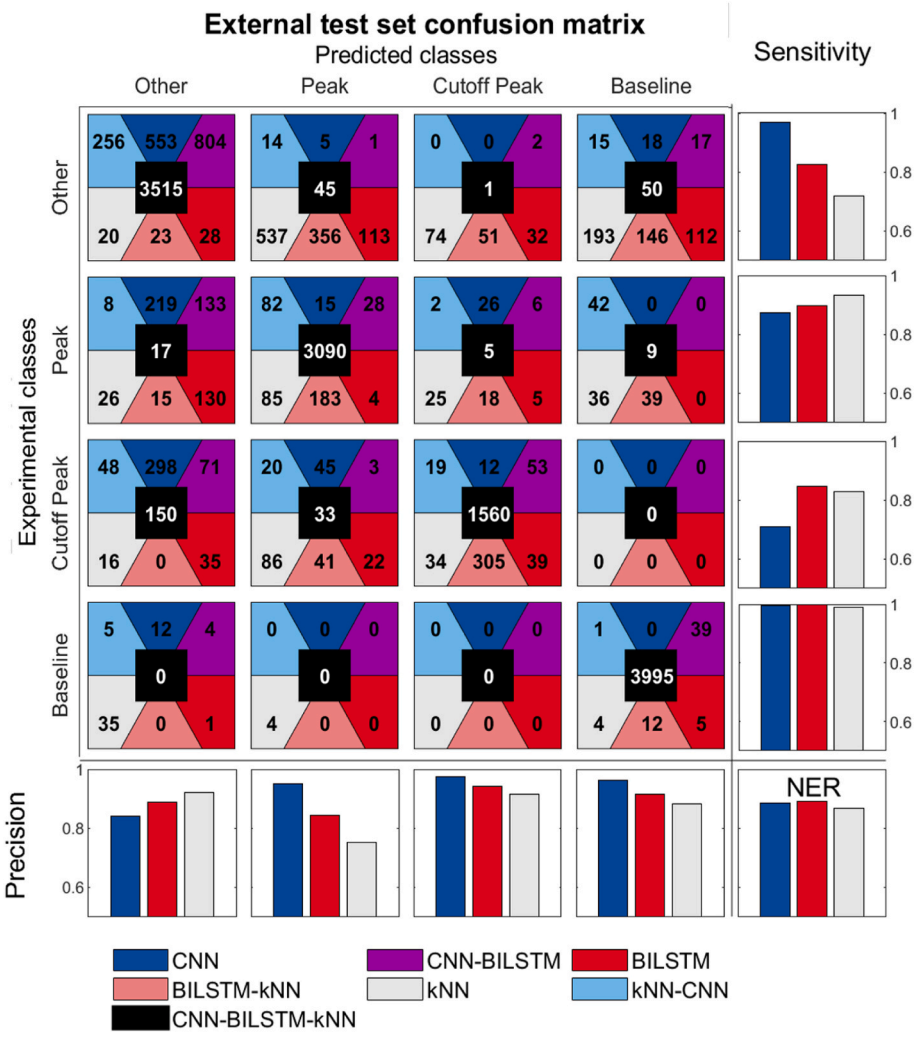


Fig. 3. Aggregated confusion plot for the three approaches (CNN, BILSTM and kNN in blue, red, and grey, respectively) calculated for the external test set. Details about construction and interpretation are given in the text (Results section). The size of the areas is proportional to the logarithm of the number of profiles. Sensitivities, precisions and NER are also reported as bar plots. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

in particular for the 'Peak' class. In this case, the kNN has the greatest sensitivity, followed by BILSTM and CNN (cell 2 × 5, Fig. 3, Table S1). This is due to the profiles labelled as 'Peak' but classified as 'Other' by the CNN and BILSTM models (blue, pink and red areas, 2 × 1 and 3 × 1 cells, Fig. 3), as discussed before. It means that the kNN model can correctly classify the highest number of profiles labelled as 'Peak' compared to BILSTM and CNN. The difference in the sensitivities for the 'Peak' class between the kNN and the CNN is 6 %. Looking at the precisions for the class 'Peak' (cell 5 × 2, Fig. 3), the trend is the opposite: the CNN has the best precision and the kNN the worst. The grey, light blue and light red areas, related to the classifications of the kNN model, show the higher values in the cells 1 × 2 and 3 × 2, indicating that kNN tends to assign to the class 'Peak' profiles belonging to different classes. In particular, the difference of precisions between the CNN and kNN is 20 %. This can be explained considering the criteria adopted during the labelling, where a little difference can discriminate between a class or another. Such small differences between different classes can be problematic to detect for a local model as kNN.

Considering the class 'Cutoff peak', the BILSTM model has the highest sensitivity and the CNN the smallest one (cell 3 × 5, Fig. 3). As for the class 'Peak', the sensitivity of the CNN model is lower compared to the other models, because it tends to classify more profiles as 'Other'. This tendency increases the precision for the CNN model which is the highest also for this class. However, the differences among the three models for this class are less evident and the performances are satisfactory for all the models.

All the three models had excellent performances considering the class 'Baseline', both in terms of sensitivities and precisions (cells 4 × 5 and 5 × 4, respectively, Fig. 3, Table S1). Looking at the aggregated confusion plot, the number of profiles classified as 'Baseline' from all the three models and actually belonging to this class is 3995 (black area, cell 4 × 4 Fig. 3 and Fig. S3) over a total number of 4056 profiles labelled as 'Baseline' in the external test set (Table 1). Thus, the 98.4% of the 'Baseline' profiles have been correctly classified by all the classification models, further indicating the excellent performances for this class.

We identified the profiles misclassified by all the models, represented by the black areas in the off-diagonal cells for a total of 310 profiles (2 % of all the profiles in the external test set). For the cells 3 × 4 (experimental class: 'Cutoff peak'; predicted: 'Baseline'), 4 × 2 (experimental class: 'Baseline'; predicted: 'Peak') and 4 × 3 (experimental class: 'Baseline'; predicted: 'Cutoff peak'), no misclassified profiles from all the models were found. In all the other cases, it can be seen that misclassifications mostly depend on borderline profiles (i.e., profiles at the edge between two classes) and by errors of the labelling phase. For instance, in the most representative group corresponding to the cell 3 × 1 (experimental class: 'Cutoff peak'; predicted: 'Other '), the profiles do

not clearly show the inflection point or the tail is slightly over the 0.1 threshold. Also it is possible to notice that some of the profiles have an expected label that is not correct, probably depending on mislabelling during the manual classification of the profiles. All the profiles are shown in Fig. S3. Extending the same trend also for the validation and the training sets, some underestimation of the classification performances can be assumed.

3.2. Computational time

We compared the computational time required by the different models to perform the classification on the 24,036 profiles of the validation set (Table 3). The calculation has been performed with an Intel® Core™ i7-6950X CPU processor with a dedicated RAM of 32 GB. The time needed by CNN and the BILSTM models are comparable, the slightly longer running time for the CNN model can be explained by the greater number of hidden layers. On the other side, the kNN model requires significantly more time compared to the other two models, i.e., it is about 20 times slower. The difference in time required by the models to perform the classification task is important in this context, considering that the analysis of a full GC-MS dataset would produce a considerably high number of profiles.

3.3. Comparison with literature model

In order to further evaluate the classification approaches, we compared the classification models to the convolutional neural network described in Ref. [15]. The comparison is based on the ROC curves for the class 'Peak' for the respective external test sets. The curves are shown in Fig. 4. Since these results are based on different test sets, the comparison is qualitative with the aim to verify the influence of the data more than the classification performance.

The ROC curve for the deep neural network considering the 'Peak' class in Ref. [15] was already next to the top left corner, nonetheless the curves for the three models trained on our data show a further improvement. The AUC reported in Ref. [15] was already high, reaching 0.95, however both the CNN and the BILSTM models have an even higher value, equal to 0.96 and 0.97, respectively, suggesting that the adopted criteria for the labelling of the profiles made possible a slight increase of the classification performances. Considering the kNN model, the AUC for the 'Peak' class is 0.94, slightly lower compared to the CNN and the BILSTM models. The AUC value for all the classes for the CNN, BILSTM and kNN models is reported in Table S1.

4. Conclusions

In this work we have developed three classification models for the shape recognition of resolved chromatographic profiles. The models have been trained with PARAFAC2 resolved profiles, which were manually labelled in four classes: 'Peak', 'Cut off peak', 'Baseline' and 'Other', according to previous works.

The performances of the models have been analysed by means of an aggregated confusion plot where the classification results of the three models have been merged, resulting in a deeper insight of the different model trends. Overall, all the three models seem effective. Nonetheless, there are hints suggesting that the two models based on deep neural networks are learning the underlying criteria applied during the labelling process from the data, while the kNN model seems less robust compared to the other two.

Overall, the analysis of the results obtained considering the external test set allowed to characterize the three models, where in general the CNN has the highest precisions and tends to classify more profiles as 'Other', the kNN has the highest sensitivities, but fails to properly classify borderline profiles and the BILSTM seems as a compromise between the other two methods. Moreover, the computational times indicate the two deep learning approach as significantly faster compared to the kNN model.

Table 3
Computational time for the classification of the internal validation set.

Model	Time (seconds)
CNN	4.34
BILSTM	2.43
kNN	64.3

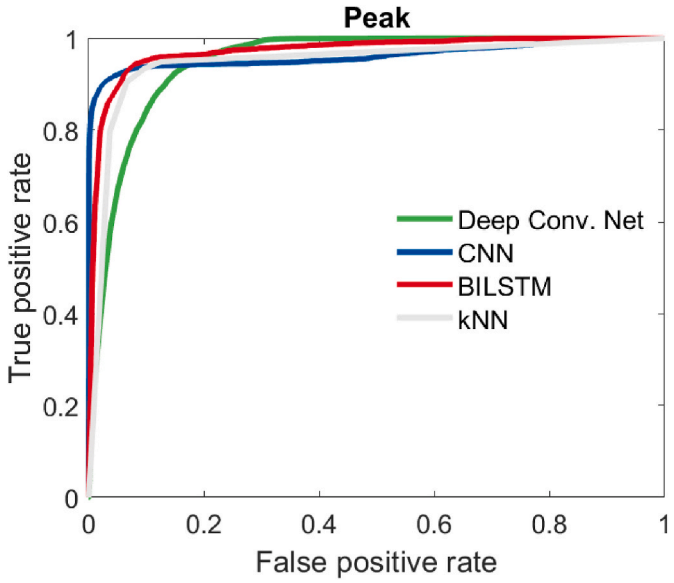


Fig. 4. Comparison of the ROC curves for the 'Peak' class of the deep convolutional net from Ref. [15] and the proposed CNN, BILSTM and kNN models. The curves are calculated on the external test set.

Credit

Conceptualization (GB, RB, CV), Data curation (GB, HY, CV), Formal analysis (GB), Software (GB), Supervision (RB, DB), Writing-original draft (GB), Review and editing (GB, HY, CV, RB, DB).

Author statement

We look forward to hearing if our paper is suitable for the special issue honoring Svante Wold.

Declaration of competing interest

None.

Data availability

Data will be shared on our home page

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2023.105002.

References

[1] F.R. Pinu, D.J. Beale, A.M. Paten, K. Kouremenos, S. Swarup, H.J. Schirra, D. Wishart, Systems biology and multi-omics integration: viewpoints from the metabolomics research community, Metabolites 9 (2019), https://doi.org/10.3390/metabo9040076.

[2] R. Aebersold, M. Mann, Mass spectrometry-based proteomics, Nature 422 (2003) 198–207, https://doi.org/10.1038/nature01511.
[3] C.S. Hsu, C.L. Hendrickson, R.P. Rodgers, A.M. McKenna, A.G. Marshall, Petroleomics: advanced molecular probe for petroleum heavy ends, J. Mass Spectrom. 46 (2011) 337–343, https://doi.org/10.1002/jms.1893.
[4] R. Tauler, H. Parastar, Big (Bio)Chemical Data Mining Using Chemometric Methods: A Need for Chemists, Angewandte Chemie International Edition, 2018, https://doi.org/10.1002/anie.201801134.
[5] A.C. Schrimpe-Rutledge, S.G. Codreanu, S.D. Sherrod, J.A. McLean, Untargeted metabolomics strategies—challenges and emerging directions, J. Am. Soc. Mass Spectrom. 27 (2016) 1897–1905, https://doi.org/10.1007/s13361-016-1469-y.
[6] L.D. Roberts, A.L. Souza, R.E. Gerszten, C.B. Clish, Targeted metabolomics, Curr. Protoc. Mol. Biol. 98 (2012), https://doi.org/10.1002/0471142727.MB3002S98, 30.2.1-30.2.24.
[7] D.C. Sévin, A. Kuehne, N. Zamboni, U. Sauer, Biological insights through nontargeted metabolomics, Curr. Opin. Biotechnol. 34 (2015) 1–8, https://doi.org/10.1016/J.COPBIO.2014.10.001.
[8] J.H. Christensen, J. Mortensen, A.B. Hansen, O. Andersen, Chromatographic preprocessing of GC–MS data for analysis of complex chemical mixtures, J. Chromatogr. A 1062 (2005) 113–123, https://doi.org/10.1016/J.CHROMA.2004.11.037.
[9] L.G. Johnsen, T. Skov, U. Houlberg, R. Bro, An automated method for baseline correction{,} peak finding and peak grouping in chromatographic data, Analyst 138 (2013) 3502–3511, https://doi.org/10.1039/C3AN36276K.
[10] A. Alonso, S. Marsal, A. Julià, Analytical methods in untargeted metabolomics: state of the art in 2015, Front. Bioeng. Biotechnol. 3 (2015) 23, https://doi.org/10.3389/FBIOE.2015.00023/BIBTEX.
[11] J.M. Amigo, T. Skov, R. Bro, ChroMATHography: solving chromatographic issues with mathematical models and intuitive graphics, Chem. Rev. 110 (2010) 4582–4605, https://doi.org/10.1021/cr900394n.
[12] R. Bro, C.A. Andersson, H.A.L. Kiers, PARAFAC2—Part II. Modeling chromatographic data with retention time shifts, J. Chemometr. 13 (1999) 295–309, https://doi.org/10.1002/(SICI)1099-128X(199905/08)13:3/4<295::AID-CEM547>3.0.CO;2-Y.
[13] I.H.M. van Stokkum, K.M. Mullen, V.V. Mihaleva, Global analysis of multiple gas chromatography-mass spectrometry (GC/MS) data sets: a method for resolution of co-eluting components with comparison to MCR-ALS, Chemometr. Intell. Lab. Syst. 95 (2009) 150–163, https://doi.org/10.1016/J.CHEMOLAB.2008.10.004.
[14] L.G. Johnsen, P.B. Skou, B. Khakimov, R. Bro, Gas chromatography – mass spectrometry data processing made easy, J. Chromatogr. A 1503 (2017) 57–64, https://doi.org/10.1016/J.CHROMA.2017.04.052.
[15] A.B. Risum, R. Bro, Using deep learning to evaluate peaks in chromatographic data, Talanta 204 (2019) 255–260, https://doi.org/10.1016/j.talanta.2019.05.053.
[16] B. Quintanilla-Casas, M. Marin, F. Guardiola, D.L. García-González, S. Barbieri, A. Bendini, T.G. Toschi, S. Vichi, A. Tres, Supporting the sensory panel to grade virgin olive oils: an in-house-validated screening tool by volatile fingerprinting and chemometrics, Foods 2020, Vol. 9, Page 1509 9 (2020) 1509, https://doi.org/10.3390/FOODS9101509.
[17] R.B. Cattell, "Parallel proportional profiles" and other principles for determining the choice of factors by rotation, Psychometrika 9 (1944) 267–283, https://doi.org/10.1007/BF02288739.
[18] H. Yu, L. Guo, M. Kharbach, W. Han, Multi-way analysis coupled with near-infrared spectroscopy in food industry: models and applications, Foods 10 (2021), https://doi.org/10.3390/foods10040802.
[19] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 1986 323:6088 323 (1986) 533–536, https://doi.org/10.1038/323533a0.
[20] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, https://doi.org/10.1162/NECO.1997.9.8.1735.
[21] M. Sundermeyer, R. Schlüter, H. Ney, LSTM Neural Networks for Language Modeling, INTERSPEECH, 2012.
[22] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997).
[23] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, 3rd international conference on learning representations, ICLR 2015 - Conf. Track Proc. (2014).
[24] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Statistician 46 (1992) 175–185, https://doi.org/10.1080/00031305.1992.10475879.
[25] D. Ballabio, F. Grisoni, R. Todeschini, Multivariate comparison of classification performance measures, Chemometr. Intell. Lab. Syst. 174 (2018) 33–44, https://doi.org/10.1016/J.CHEMOLAB.2017.12.004.
[26] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (2006) 861–874, https://doi.org/10.1016/J.PATREC.2005.10.010.
[27] J.E. Cohen, R. Bro, Nonnegative PARAFAC2: a flexible coupling approach, 10891, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, 2018, https://doi.org/10.1007/978-3-319-93764-9_9, 89–98.
[28] Davide Ballabio, Viviana Consonni, Classification tools in chemistry. Part 1: linear models. PLS-DA, Anal. Methods 5 (2013) 3790–3798, https://doi.org/10.1039/C3AY40582F.