Analytica Chimica Acta 1258 (2023) 341147

Contents lists available at ScienceDirect

Analytica Chimica Acta

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

Non-linear shrinking of linear model errors

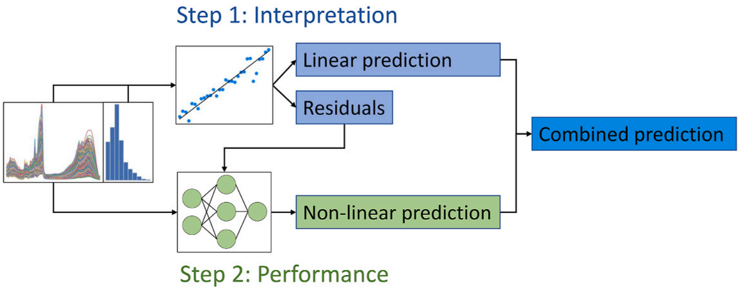
Runar Helin [*], Ulf Indahl, Oliver Tomic, Kristian Hovde Liland

Norwegian University of Life Sciences, Faculty of Science and Technology, Ås, Norway

HIGHLIGHTS

- Residual shrinking is developed for both regression and classification problems.
- The proposed strategy can improve predictions while retaining interpretability.
- This contributes to explainable AI by shrinking the black box of ANNs.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Residual modelling
Hybrid model
Neural network
Interpretation
Deep learning
PLSR

ABSTRACT

Background: Artificial neural networks (ANNs) can be a powerful tool for spectroscopic data analysis. Their ability to detect and model complex relations in the data may lead to outstanding predictive capabilities, but the predictions themselves are difficult to interpret due to the lack of understanding of the black box ANN models. ANNs and linear methods can be combined by first fitting a linear model to the data followed by a non-linear fitting of the linear model residuals using an ANN. This paper explores the use of residual modelling in high-dimensional data using modern neural network architectures.

Results: By combining linear- and ANN modelling, we demonstrate that it is possible to achieve both good model performance while retaining interpretations from the linear part of the model. The proposed residual modelling approach is evaluated on four high-dimensional datasets, representing two regression and two classification problems. Additionally, a demonstration of possible interpretation techniques are included for all datasets. The study concludes that if the modelling problem contains sufficiently complex data (i.e., non-linearities), the residual modelling can in fact improve the performance of a linear model and achieve similar performance as pure ANN models while retaining valuable interpretations for a large proportion of the variance accounted for.

Significance and novelty: The paper presents a residual modelling scheme using modern neural network architectures. Furthermore, two novel extensions of residual modelling for classification tasks are proposed. The study is seen as a step towards explainable AI, with the aim of making data modelling using artificial neural networks more transparent.

[*] Corresponding author.
E-mail address: runar.helin@nmbu.no (R. Helin).

https://doi.org/10.1016/j.aca.2023.341147
Received 21 December 2022; Received in revised form 17 March 2023; Accepted 24 March 2023
Available online 4 April 2023

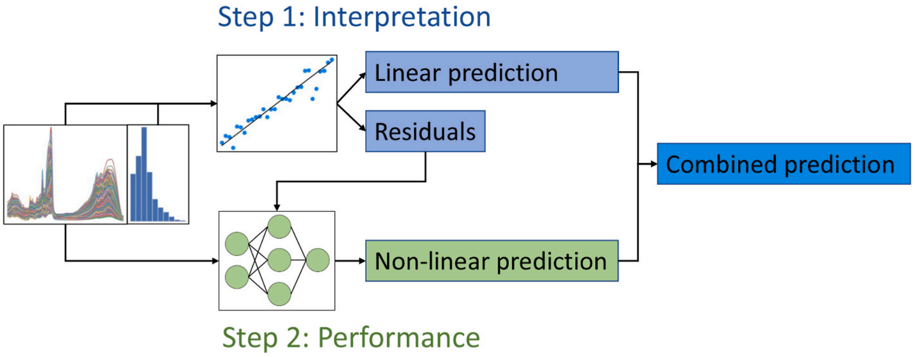


Fig. 1. Schematic illustration of the non-linear residual modelling. The input data is processed by a PLS model to generate interpretable linear prediction $\widehat{\mathbf{y}}_{\mathrm{PLS}}$. The linear prediction is subtracted from the true response to obtain the residuals $\mathbf{r}_{\mathrm{PLS}}$ which are modelled by the ANN for improved performance of the hybrid model. The final prediction $\widehat{\mathbf{y}}$ of the hybrid model is obtained as the sum of $\widehat{\mathbf{y}}_{\mathrm{PLS}}$ and $\widehat{\mathbf{r}}_{\mathrm{ANN}}$.

1. Introduction

Partial Least Squares (PLS) [1,2] is a popular method for linear regression model building and data analysis of high-dimensional spectroscopic and multicollinear data. The linear PLS models are considered to be relatively simple, and interpretable by considering the score- and loading vectors obtained from the model building process. In light of Beer Lambert's law, which states that the absorbance of a chemical species is proportional to the concentration of that species in a sample, linear models are usually assumed to be sufficient for modelling based on spectroscopic data modelling. Although often valid, effects caused by physical or chemical interferents, scattering effects or complex structured materials are sources that may violate Beer Lambert's law [3]. Recent studies have shown that non-linear models based on artificial neural networks (ANNs) can sometimes achieve better prediction performance than PLS models [4–6]. These findings lend credibility to the inclusion of neural networks in the standard toolbox for spectroscopic data analysis.

The rapid advances in deep learning technology have led to an increased interest in the study of neural networks and their potential for model building based on spectroscopic data. Despite this increased interest, research on problems related to image- and text analysis are still the dominant areas of application. By adapting techniques from established deep learning models, it has been demonstrated that convolutional neural networks (CNNs) are especially effective on 1D spectroscopic data [7–11]. Furthermore, as CNNs have achieved high prediction performance even without including explicit preprocessing of the raw spectra [5,12], there are indications that the effect of the preprocessing step is something that can actually be learned during the CNN learning process, given that relevant variation in the interfering signals is present in the spectra.

The ability to learn good feature representations automatically from the data is one of the strongest assets of ANN models. However, this comes at the cost of model transparency and makes the resulting neural network model weaker in terms of interpretations when compared to the linear models obtained by PLS and related methods. The troublesome understanding of ANN models is caused by the complex sequences of non-linear transformations for calculating the network outputs (predictions) from the input data.

In order to obtain useful interpretations along with the modelling there are two obvious alternatives: 1) One can discard ANN-based models and rely solely on interpretable models obtained by simpler (linear) strategies at the possible cost of some predictive performance. 2) One can focus on developing supplementary methodology to obtain more useful interpretations from ANN models, such as feature importances [13]. A third alternative is to consider a hybrid modelling approach that splits the model building process into a linear part (for interpretations) and a non-linear part (for enhanced model performance). In the literature, the latter approach has different names (both hybrid modelling and residual modelling have been used).

1.1. Comparison to the literature

Hybrid and residual modelling have been proposed in various forms. A popular approach to hybrid modelling is to use an ANN as a feature extractor [18]. This procedure can be used to extract non-linear features to later be modelled by other models, such as a PLS model [14]. More recently, large ANNs pre-trained on massive datasets of images are used to extract general features from images to be used as input for other models [19]. Another form of hybrid modelling is to use linear models to enrich the features fed to an ANN [15]. This is a kind of feature engineering approach to help the ANN model find useful relations in the data, and can be useful in situations with little available data. A third approach, and the one focused on in this study, is to use an ANN to model the residuals from a linear model [16]. The concept was first proposed to improve prediction performance of an ANN which suffered from poor generalisations [17]. The performance was improved by training the ANN to learn the difference between the linear model prediction and the response value and subsequently combine the linear and ANN predictions. Residual modelling has also been successfully applied to time series analysis based on the linear autoregressive integrated moving average (ARIMA) [20–22].

With increasing amounts of data collected and better deep learning technology, modern ANN models are often outperforming more traditional models. Thus, the residual modelling might not necessarily improve the prediction performance as was shown in the earlier studies. A more interesting avenue is therefore to use the residual modelling framework to better understand the problem and data. The present study explores this possibility of interpretation within the residual modelling framework. The framework is similar to Hussain et al. [17], but uses modern deep learning techniques such as rectified linear units (ReLU) activation functions and dropout layers [23] to obtain better performing ANN models. Different from previous studies, we apply modern ANN architectures to explore the effectiveness of non-linear modelling of the residuals from a linear model with a focus on maintaining model interpretability. The study focuses on high-dimensional spectroscopic data which has, to our knowledge, not previously been used with residual modelling. PLS and CNN models are used as examples of linear and deep learning models, respectively, because of their standing in the literature regarding interpretation possibilities and performance with spectral data [24,25]. Fig. 1 shows a basic illustration of the modelling scheme. Prediction of a new sample is obtained by adding the predicted residual term and the prediction from the linear model together.

In our study, the residual modelling is applied to four different high-dimensional datasets. We also present two novel approaches that extend the idea of doing residual modelling also for classification problems. The

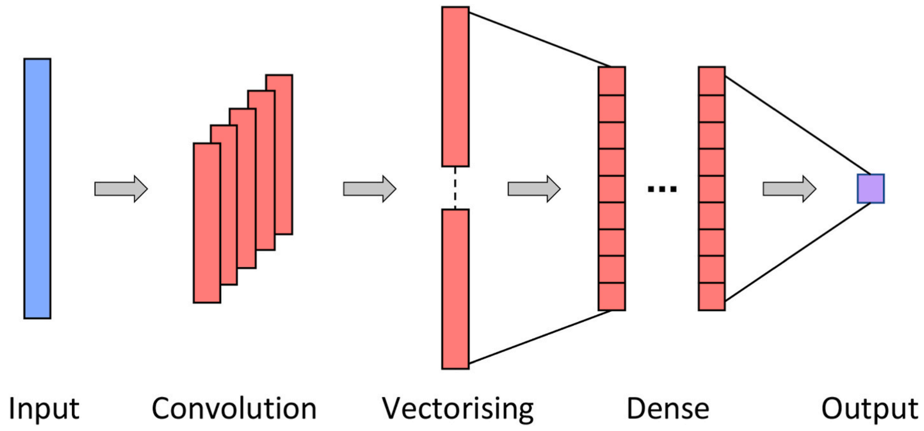


Fig. 2. Illustration of the base neural network architecture used for the 1D datasets.

predictive performance of the hybrid modelling scheme is compared to both pure PLS modelling and classical ANN modelling. Furthermore, possible diagnostic tools for model interpretations are presented and discussed for all four problems.

2. Methods and material

In the following, we give an overview of the models used in this study and a detailed description of the non-linear residual modelling approach.

2.1. Models

Predictive model building is a learning process for mapping data from some input space $\mathbb{R}^p$ to an output space $\mathbb{R}^c$. For the regression problems considered here, the output space is one-dimensional ($c = 1$) and for classification problems the number of dimensions c equals the number of groups.

In the following, let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a data matrix with n measurements of p-dimensional samples. Assume that $\mathbf{X}$ is centred and possibly further preprocessed. Let $\mathbf{x}^T$ denote a row-vector from this matrix with corresponding response value(s) $\mathbf{y} \in \mathbb{R}^c$.

The prediction mapping of a linear model can be described by the equation

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{c \times p}$ and $\mathbf{b} \in \mathbb{R}^c$ are the parameters called *weights* and *biases*, respectively, in machine learning and *regression coefficients* and *intercepts* in statistics and chemometrics. In the case of a PLS model, equation (1) becomes $\hat{\mathbf{y}} = \boldsymbol{\beta}\mathbf{x} + \bar{\mathbf{y}}$, where $\mathbf{W} = \boldsymbol{\beta}$ is a matrix with regression coefficients found through the PLS algorithm and $\bar{\mathbf{y}}$ are the mean response values over the training set. The PLS model projects the input data onto a low-dimensional latent space represented by score vectors $\mathbf{t}_a$. These score vectors are found sequentially as the linear combinations of the original features that maximise the covariance between an input matrix $\mathbf{X}$ and response $\mathbf{y}$. In the PLS model building, the score vectors are associated with corresponding loading vectors $\mathbf{p}_a$ which represent coordinates of the features in the compressed subspace. Together, the score- and loading vectors form rank one matrices which added together form the original data (maximum number of components) or an approximation: $\hat{\mathbf{X}} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T$.

An ANN model is essentially a function that maps an input vector to some scalar or vector output through a sequence of non-linear transformations $f_1, \ldots, f_D$. The model is trained by adjusting the network parameters (weights) in a supervised fashion. For a network with layers $d = 1, \ldots, D$, each containing n_d nodes, the prediction of a sample $\mathbf{x}^T$ is obtained by the composed mapping

$$\hat{\mathbf{y}} = (f_D \circ \cdots \circ f_1)(\mathbf{x}^T). \tag{2}$$

Each transformation is represented by a non-linear function (activation function) of the weighted sum of its vector input values, i.e., a non-linear vector-to-vector mapping associated with the nodes and weights between two *layers* of the network architecture. For example, the transformation associated with a fully connected layer d is defined as $f_d(\mathbf{z}^T) := \varphi(\mathbf{W}_d\mathbf{z} + \mathbf{b}_d)$, where $\varphi(\cdot)$ is the non-linear activation function operating element-wise on the argument vector. The weight matrix $\mathbf{W}_d$ can alternatively be replaced by a convolution matrix/operator to obtain a convolution layer. Each layer is associated with its own set of weights and biases that during the training process are adjusted iteratively by an error back-propagation gradient descent algorithm.

The most important goal of regression modelling is to identify model parameter values resulting in small prediction errors when applying the model. A popular strategy is to search for the model parameters minimising the mean squared error (MSE) of the training data: $\| \mathbf{y} - \hat{\mathbf{y}} \|^2 / n$, where n is the number of samples and $\hat{\mathbf{y}}$ are the associated model predictions.

In PLS regression, the objective implemented by the algorithm includes decomposition and dimension reduction of the data matrix $\mathbf{X}$ followed by linear least squares modelling with the reduced data. For a neural network the objective is chosen as a loss function (cost function) to be minimised by a gradient descent search. In classification problems, the objective is to minimise the number of misclassified samples. For a classification problem including c different classes (groups), and an n-dimensional vector representing the class labelling, the latter is usually replaced by a (one-hot encoded) $\mathbf{Y} \in \mathbb{R}^{(n \times c)}$ dummy matrix.

In the PLS modelling approach to classification the dummy matrix is taken as the response data to obtain the model predictions $\hat{\mathbf{Y}}$. Thereafter, a classification model can be obtained by linear discriminant analysis (LDA) [26] of the fitted values $\hat{\mathbf{Y}}$.

The dummy matrix $\mathbf{Y}$ can also be taken as the responses for training an ANN model with c output nodes including softmax transformations of the model output to produce class membership probabilities. A categorical cross-entropy loss function, defined by

$$L(\mathbf{Y}, \hat{\mathbf{Y}}) = -\sum_{i=1}^{n} \sum_{j=1}^{c} Y_{i,j} \ln \hat{Y}_{i,j} \tag{3}$$

is most often used for training the classification ANN, where $\hat{Y}_{i,j}$ is the predicted probability of sample i to belong to class j.

2.2. Neural network architecture and training

Choosing an efficient neural network architecture is challenging

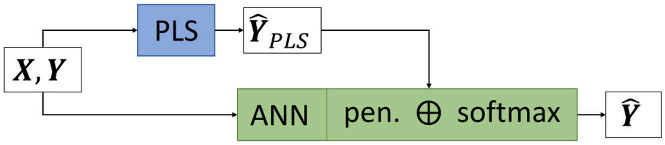


Fig. 3. Illustration of the CE-shrinking approach. The penalised contribution of the ANN prediction and the PLS prediction are added together (element-wise) before the softmax activation in the network. Note that the PLS prediction is unchanged during the ANN training phase.

given the almost unlimited number of possible network configurations. One of the most important aspects is to find the correct model complexity/capacity. The network must be sufficiently complex to model the data. However, inclusion of too many weights may lead to model overfitting. Due to limited research on network architectures for 1D spectroscopic data, there are no clear recommendations for the choice of model architectures, often resulting in a large amount of time spent on model tuning.

In the present paper, our choice(s) of architecture(s) are based on prior experience reported in the literature [12,27]. An illustration is shown in Fig. 2.

This architecture includes one convolution layer followed by a stack of fully connected (dense) layers. A non-linear activation function is used element-wise in the mappings between each layer, and the dropout principle [23] is used for the mappings into the final layer. The number and size of the convolution filters, the number of fully connected layers, the number of nodes in each hidden layer, the type of activation function and the dropout rate are all to be considered as hyperparameters.

The hyperparameters control the complexity of the model and are tuned to each specific dataset in the present work using a Bayesian optimisation [28]. A Python algorithm utilising the Keras library of TensorFlow [29] implements the training process. Specifications of the ranges used in the hyperparameter search are provided in the Supplementary materials. The computations were executed on an NVIDIA Quadro RTX 8000 graphics processing unit (GPU). The PLS and LDA modelling were done using Python and the scikit-learn package version 1.1.1 [30].

The ANNs used for model predictions and non-linear residual modelling on 1D data use the base architecture described above. Among the datasets there is one set containing 2D images, for which the ANN used is a standard 2D CNN suitable for image recognition. This 2D architecture consists of blocks of convolution layers and batch normalisation layers with one fully connected layer and regularising dropout before the output layer. Dimension reduction is obtained by strided convolutions between some network layers. For more details on the architecture, see the Supplementary materials.

The ANN training is based on model updating during multiple epochs of error back-propagation. The number of epochs (cycles through the training data) is essential for tuning the networks as it affects the amount of over/under fitting. Determination of the appropriate number of epochs for training is usually done by monitoring the loss (prediction error) on an independent validation set during the training phase.

2.3. Non-linear residual modelling

The concept behind residual modelling is to improve the prediction performance of a linear model by explicitly modelling the prediction errors (residuals) from that linear model using a neural network. This idea is most straight-forward for regression problems, but the same concept can be applied on classification problems (described below). The residual modelling procedure is illustrated in Fig. 1. First, the input data ($\mathbf{X}$, $\mathbf{y}$) are used to train a PLS model which generates the linear prediction $\widehat{\mathbf{y}}_{\text{PLS}}$. Second, an ANN is trained using the residuals $\mathbf{r}_{\text{PLS}} = \mathbf{y} - \widehat{\mathbf{y}}_{\text{PLS}}$ as target for the network. Note that the network takes the same data $\mathbf{X}$ as input. Denoting the prediction of the residuals $\widehat{\mathbf{r}}_{\text{ANN}}$, the final prediction of the non-linear residual shrinking problem is the sum $\widehat{\mathbf{y}} = \widehat{\mathbf{y}}_{\text{PLS}} + \widehat{\mathbf{r}}_{\text{ANN}}$.

2.3.1. Classification problems

It is not as obvious how to transform the residual modelling from minimising continuous errors to correcting categorical misclassifications. We explore and compare two different approaches called MSE-shrinking and CE-shrinking (described below) representing different ways to utilise the ANN to correct the linear model outputs.

In the MSE-shrinking approach, the ANN is trained to learn the matrix of residuals from the multi-class PLS prediction $\mathbf{R} = \mathbf{Y} - \widehat{\mathbf{Y}}_{\text{PLS}}$. This approach is similar to regression problems where the network tries to learn the difference between the one-hot encoded matrix and the PLS prediction. The final prediction is obtained by an LDA model trained on the sum $\widehat{\mathbf{Y}}_{\text{PLS}} + \widehat{\mathbf{R}}_{\text{ANN}}$. The loss function of this network is MSE, hence the name MSE-shrinking.

In the MSE-shrinking approach, the ANN is essentially doing a regression and is unaware of the final goal of classification. In order to obtain an objective more relevant for classification problems, we also considered the CE-shrinking alternative where the network is using $\widehat{\mathbf{Y}}_{\text{PLS}}$ as additional inputs. The purpose of this approach is to try to learn the residuals implicitly by forcing the network to improve on the provided PLS predictions. This is achieved by adding the PLS prediction $\widehat{\mathbf{Y}}_{\text{PLS}}$ to the output of the last layer prior to the softmax activation as illustrated in Fig. 3. Since the loss function of this network is the commonly used categorical cross-entropy, we refer to this alternative residual modelling approach as CE-shrinking.

Upon testing, it turned out that the networks required some constraints to avoid ignoring the provided PLS predictions. Without constraints, the network models had a tendency to make the outputs of the ANN block shown in Fig. 3 extremely large. In effect this essentially made the provided PLS predictions irrelevant.

To overcome this problem, we included an L2-regularisation to the ANN contribution by adding $\lambda \parallel \mathbf{Z} \parallel$ to the loss function, where $\mathbf{Z}$ is the pre-softmax output of the ANN block. By this, the values are forced to be of a magnitude similar to the PLS predictions. Through visual inspection, it was found that $\lambda = 1$ constrained the network modelling sufficiently but not too much. Since the $\widehat{\mathbf{y}}_{\text{PLS}}$ will have values close to the range 0–1, it is reasonable to assume that $\lambda = 1$ is a good choice regardless of the dataset. An alternative to visual inspection is to compare the Frobenius norms of $\widehat{\mathbf{Y}}_{\text{PLS}}$ and $\mathbf{Z}$ to determine the appropriate value of λ. The parameter λ acts as an additional hyper-parameter that control the trade-off between interpretation and performance. With insufficient regularisation (λ chosen to be too small), the PLS model interpretations of our hybrid modelling approach may be invalid.

2.3.2. Model selection and validation

The non-linear residual modelling approach was tested on four high-dimensional benchmark datasets described in the next section. During the analysis, all datasets, except the MNIST data were divided into training (75%) and test (25%) sets in a stratified manner where applicable. The test sets were held aside for the final model evaluations. The MNIST dataset is an exception since it comes with predefined sets of training- and test data.

The regression problems were evaluated using the R^2 metric and the classification problems by prediction accuracy (i.e., percentage of correctly classified samples). During training of the ANNs, the MSE was minimised.

2.3.2.1. Model selection. Both the PLS and ANN models require tuning of model complexity. This complexity was tuned for each individual dataset. For the PLS model, the optimal number of components was found using a 5-fold cross validation on the training partition of the data. The optimal network architecture for each problem was found using the Bayesian optimisation framework described earlier, with 1/3 of the

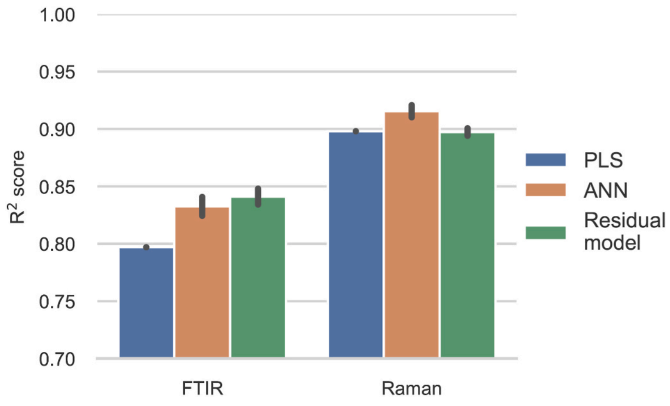


Fig. 4. Regression problems. ANN results are the means of 10 trials with standard deviation as the error bar. The y-axis is truncated.

training data as a validation set. Using a single partition of the training set reduces computational cost drastically compared to a full cross validation procedure. We have taken the point of view that a model predicting well on the validation data is also expected to perform well on the test set.

2.3.2.2. Model evaluation. During evaluation of the models, the PLS model was trained on the whole training set (75% of the total amount of data) and evaluated on the test set. The ANN model was trained on 2/3 of the training data where the final 1/3 was used as a validation set to determine how many epochs to train the ANN before convergence. The trained network was then used to evaluate the test set. Another practice included in our experiments was to do the test set evaluation based on the average of 10 neural network models using different random weight initialisations in order to obtain more robust estimates of the expected predictive performance.

Fitting the models in the non-linear residual modelling, had to be done with care. The PLS model training residuals will typically be smaller in magnitude than the validation residuals as the same samples are used for training and prediction. This means the ANN would be trained on unrealistic residual magnitudes if applied directly to the training residuals. A 5-fold cross-validation was therefore used to obtain more realistic estimates of the residuals from the PLS model before training the ANN using the cross-validated residuals as responses. After training the residual ANN on the cross-validated residuals, again using 1/3 of the data to determine convergence of the network, the test set predictions were made by first retraining the PLS model on the whole training dataset to obtain the linear test set predictions $\widehat{\mathbf{y}}_{\text{PLS}-\text{test}}$. Thereafter, the ANN model trained on the cross-validated residuals was used to obtain the non-linear predictions $\widehat{\mathbf{r}}_{\text{ANN}-\text{test}}$ of the test set.

2.4. Datasets

2.4.1. FTIR

The first dataset is for regression and contains Fourier-Transform Infrared (FTIR) spectra obtained from enzymatic protein hydrolysis processes of different rest raw materials from food production [31]. The raw materials come from fish and poultry, such as fish heads and chicken mechanical deboning residue, which were hydrolysed by different kinds of enzymes. As the response, the average molecular weight (AMW) of proteins was used, which works as a proxy for the degree of hydrolysis. In total, the dataset contains 885 spectra with a varying number of replicates. In the subsequent analysis, the spectra with wavenumbers between 1800 cm^{-1} and 700 cm^{-1} are used, resulting in 571 features for each spectrum. Prior to the modelling, the spectra were preprocessed using Savitzky-Golay smoothed second derivatives with filter width of 11 points and polynomial smoothing of 3rd degree followed by extended multiplicative signal correction (EMSC) with 2nd degree polynomial baseline correction. As an extra preprocessing step for the neural networks (both ANN modelling and residual shrinking), the data was standardised column-wise (autoscaling).

2.4.2. Raman

The second dataset is for regression and contains Raman spectra from samples of milk [32]. The dataset contains 2682 spectra with varying numbers of replicates. The wavenumber range is between 3100 cm^{-1} to 120 cm^{-1}, resulting in 2979 features for each spectrum. Each spectrum was preprocessed using EMSC with 6th order polynomial baseline correction. No further preprocessing was done prior to the neural networks for this dataset.

2.4.3. NIR

The third dataset is a classification problem and contains Near Infrared (NIR) spectra from a remote sensing hyperspectral image over an area covering 16 different vegetation and soil types in Salinas Valley, California [33]. The spectra cover the wavelength range 400 nm–2500 nm and have 204 features each. In our experiment, a random subset of 200 samples per class was used. Prior to the analysis, the hyperspectral bands corresponding to water absorption were removed. As a preprocessing step, the Standard Normal Variate (SNV) was performed. No further preprocessing was done prior to the neural networks.

2.4.4. MNIST

The final dataset is the Modified National Institute of Standards and Technology database (MNIST) dataset consisting of 28 × 28 pixel greyscale images of handwritten digits [34] used for classification. For the PLS model, we convert the samples to 1D by treating each pixel as a feature, resulting in 784 features in total. This dataset contains 70 000 samples (60 000 for training and 10 000 for testing). The classes are relatively balanced with the smallest and largest classes containing 9.02% and 11.25% of the total data respectively. Each feature was normalised to have values between 0 and 1. No further preprocessing was made for any of the models. A 2D CNN is better suited to capture features in the images since it also makes use of the spatial information in the data. Therefore, a 2D CNN was applied with the original input data shape of 28 × 28 pixels for both ANN modelling and residual shrinking.

3. Results

The prediction performance of each model is summarised in Figs. 4 and 5. Starting with the regression problems and the FTIR dataset, the best performance was achieved by the non-linear residual model with an R^2 score of 0.841. The ANN was slightly (but not significantly) worse with an R^2 score of 0.833. Both alternatives clearly outperformed the PLS model which got an R^2 score of 0.797. With the Raman dataset, the pure ANN performed slightly better than the PLS with R^2 scores of 0.915 and 0.898 respectively. In contrast to the FTIR dataset, the non-linear residual modelling did not improve on the linear prediction. The PLS models for the FTIR and Raman datasets included 28 and 16 components, respectively.

For the classification datasets we used prediction accuracy (proportion correctly classified) as performance measure since the datasets were relatively balanced. For the NIR data, the PLS model achieved an accuracy of 0.958. The ANN had the lowest accuracy of 0.955 while the MSE- and CE-shrinking methods had accuracies of 0.958 and 0.963, respectively. Taking the uncertainties into account, one cannot claim that any of the neural network based methods outperforms the PLS on this dataset, meaning that the PLS model was sufficient for this task. On the MNIST dataset, the PLS model got an accuracy of 0.877 while the pure ANN got an accuracy of 0.992. The MSE-shrinking and CE-shrinking techniques got accuracies of 0.994 and 0.986, respectively, both significant improvements over the underlying PLS model. The PLS

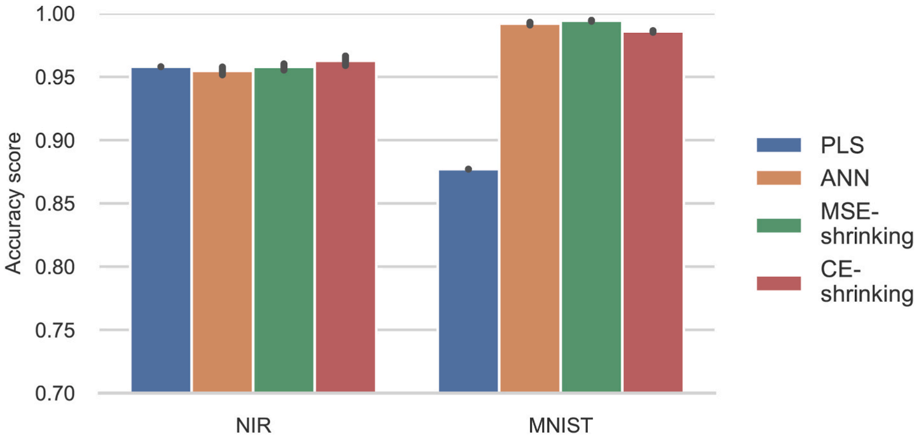


Fig. 5. Classification problems. ANN results are the means of 10 trials with standard deviation as the error bar. The y-axis is truncated.

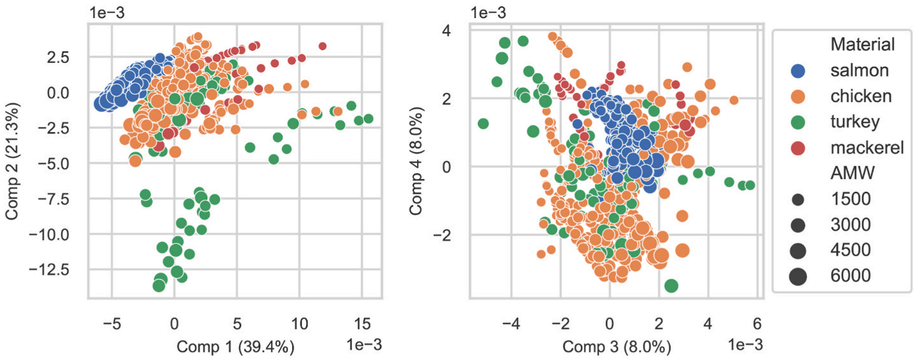


Fig. 6. Score plots from the PLS model of the FTIR dataset. Explained feature variance is shown in parentheses, raw material are encoded as colours and response values as circle sizes. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

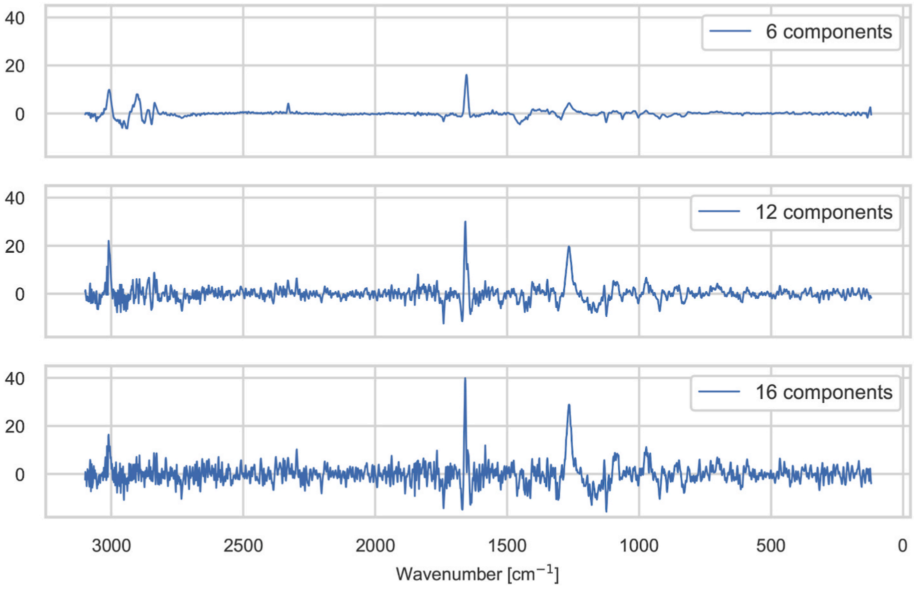


Fig. 7. Regression coefficients from PLS models trained in the Raman milk dataset using 6, 12 and 16 components.

models for the NIR and MNIST datasets used 20 and 27 components, respectively. The relatively high number of components needed must be seen in light of the large number of classes (16 for the NIR data) and the high intra-class variation (various ways of writing the same digits).

3.1. *Interpretation*

The main motivation for applying non-linear residual modelling was to utilise the power of neural networks to model non-linear relationships, while retaining the interpretability. Since the residual modelling

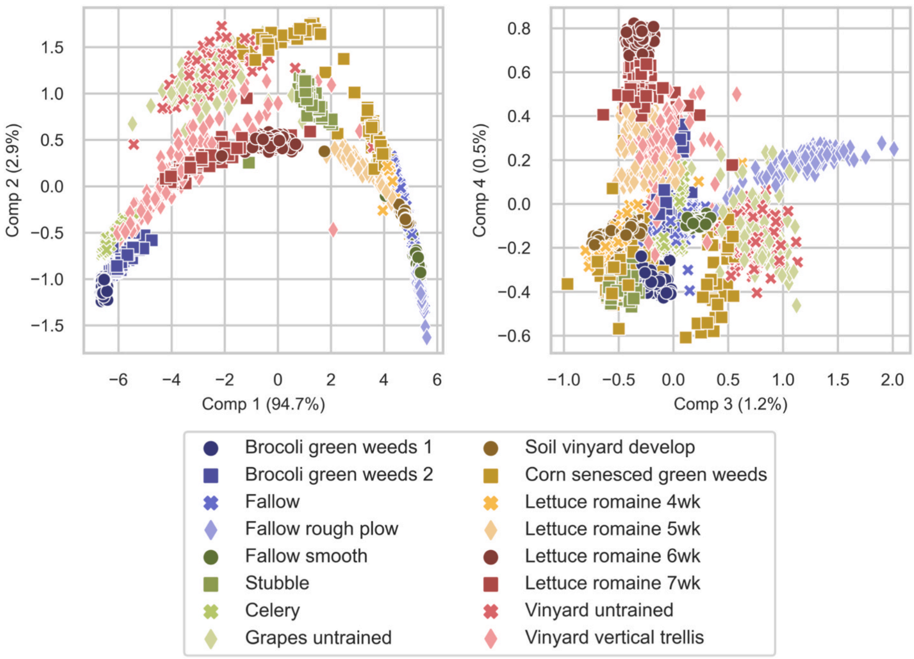


Fig. 8. Score plots from the PLS model of the remote sensing NIR dataset. Explained feature variance is shown in parentheses and symbol usage indicates different classes.

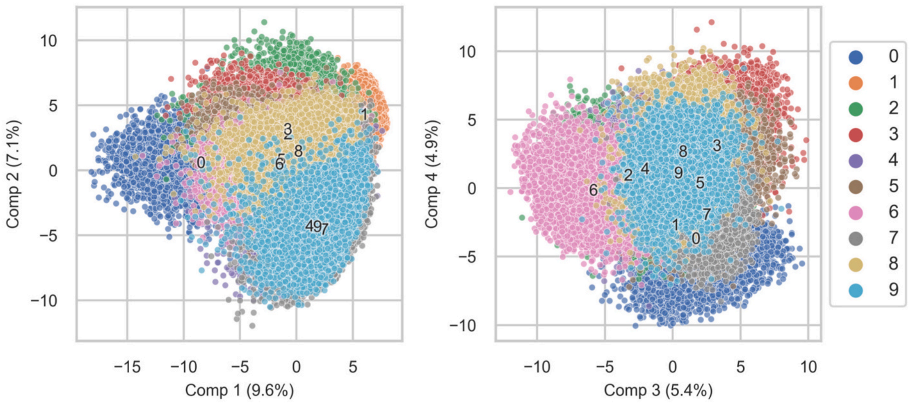


Fig. 9. Score plots from the PLS model trained on the MNIST dataset. Explained feature variance is shown in parentheses, and symbol colours indicate digits. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

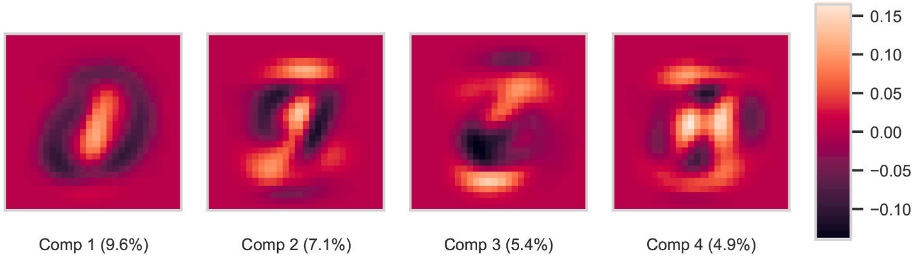


Fig. 10. Loading weights of the first four PLS components from the MNIST dataset, reshaped back to original image size (28 × 28).

does not affect the linear prediction part, all interpretations of the PLS model are still to be considered as valid. In the following we explore a selection of possible graphical diagnostic tools and their purpose. Many of these diagnostic tools are not available in pure ANN modelling and highlights the usefulness of the residual modelling approach.

Fig. 6 shows score plots of the first four PLS components from the model trained on the FTIR dataset. This kind of plot highlights patterns in the latent space, shown by similarities or clusters of samples in the scatter plots. Based on the observed groupings, it is sometimes possible to explain which samples share chemical or physical properties in the corresponding components. For instance, by colouring the samples in this dataset by their product group (chicken, turkey, mackerel and

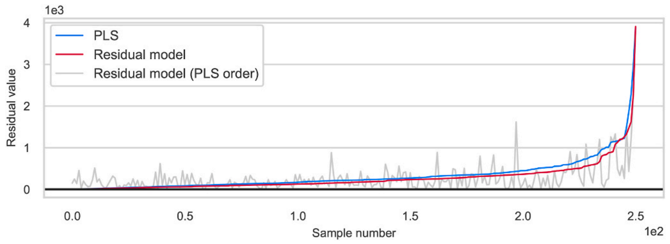


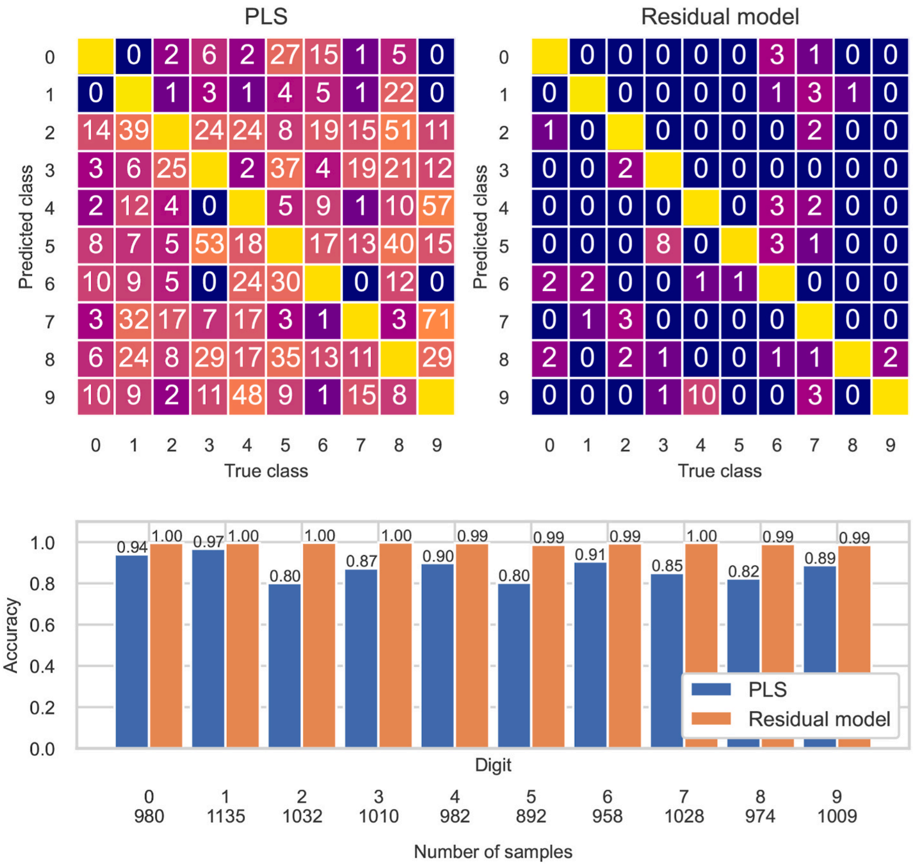
Fig. 11. Line plot of the test set residuals (absolute values) from the FTIR hydrolysates dataset for the pure PLS model and after the residual modelling. The blue and red lines show the PLS model residuals and the new residuals after the residual shrinking respectively in increasing order from left to right. The grey line shows the new residuals after the residual shrinking in the same order as the sorted PLS residuals. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

salmon), it is confirmed that the first two components distinguishes salmon from other samples. The plot also emphasises a group of turkey samples with larger AMW values. This demonstrates that from such score plots one can detect possibly unknown groupings or confirm already known or hypothesised groupings in the data. Furthermore, such plots can also be useful for detection outliers.

Fig. 7 shows line plots of the regression coefficients obtained from three different PLS models including 6, 12 and 16 PLS components, respectively, all trained on the Raman milk dataset. Such plots are useful for detecting important regions in the feature space for predicting the response, indicated by positive or negative peaks and contrasts. The model including 6 PLS components shows a peak in the region around the 1700 cm^{-1} shift. This peak is typical in systems with rich lipid content such as these samples [32] and can be an indication of varying lipid contents along with the iodine value variation. Other areas of interest for similar substances include 3000 cm^{-1} and 1300 cm^{-1}, where indeed peaks are also observed. However, more than 6 PLS components are needed before these become visually apparent. Another observation is that the regression coefficients get more noisy with an increased number of components and is an indication of model over fitting.

For the NIR remote sensing dataset, the score plots shown in Fig. 8 reveal important insights. The first two components (left figure) displays a curvy structure which indicates that there may be some effect in the spectra picked up by the model that should have been corrected for in the preprocessing. A hypothesis for explaining this phenomenon is that it has something to do with the pre-treatment of removing the water-absorbance part of the spectra prior to the analysis. In both plots we get a visual confirmation of the similarity of samples within the same category as well as indication of outlying samples, e.g., the Vinyard vertical trellis observation at around coordinates (2,-0.5) in the leftmost plot.

For the MNIST dataset, the associated score plots in Fig. 9 show how similarities between the different classes can be revealed. The score plot of the first two components, shows that digits with similar appearance such as 4, 9 and 7 tend to be grouped closely together. Furthermore, samples of the digit 1 lie far away from samples of digit 0, indicating that the first component distinguishes curves from straight lines. Another interesting observation is that the amount of explained variance is fairly low for the earlier components compared to the models for the other datasets, meaning that PLS modelling requires more components to effectively capture all variance of the MNIST features. Since the MNIST dataset consists of 2D sample images, it is also possible to visualise the PLS loading weights as images in the original 28 × 28 pixel image space as shown in Fig. 10. The loading weights highlight regions in the input space with large impact on the resulting PLS model(s) and associated


PLS

Predicted class \ True class	0	1	2	3	4	5	6	7	8	9
0		0	2	6	2	27	15	1	5	0
1	0		1	3	1	4	5	1	22	0
2	14	39		24	24	8	19	15	51	11
3	3	6	25		2	37	4	19	21	12
4	2	12	4	0		5	9	1	10	57
5	8	7	5	53	18		17	13	40	15
6	10	9	5	0	24	30		0	12	0
7	3	32	17	7	17	3	1		3	71
8	6	24	8	29	17	35	13	11		29
9	10	9	2	11	48	9	1	15	8	

Residual model

Predicted class \ True class	0	1	2	3	4	5	6	7	8	9
0		0	0	0	0	0	3	1	0	0
1	0		0	0	0	0	1	3	1	0
2	1	0		0	0	0	0	2	0	0
3	0	0	2		0	0	0	0	0	0
4	0	0	0	0		0	3	2	0	0
5	0	0	0	8	0		3	1	0	0
6	2	2	0	0	1	1		0	0	0
7	0	1	3	0	0	0	0		0	0
8	2	0	2	1	0	0	1	1		2
9	0	0	0	1	10	0	0	3	0	

Fig. 12. Confusion matrices of test set predictions on the MNIST dataset. Left: PLS model, right: PLS + non-linear residual modelling. The numbers in the diagonals are removed for clarity and the information is summarised in the bar plot at the bottom. The bar plot shows the classification accuracy per class for each model.

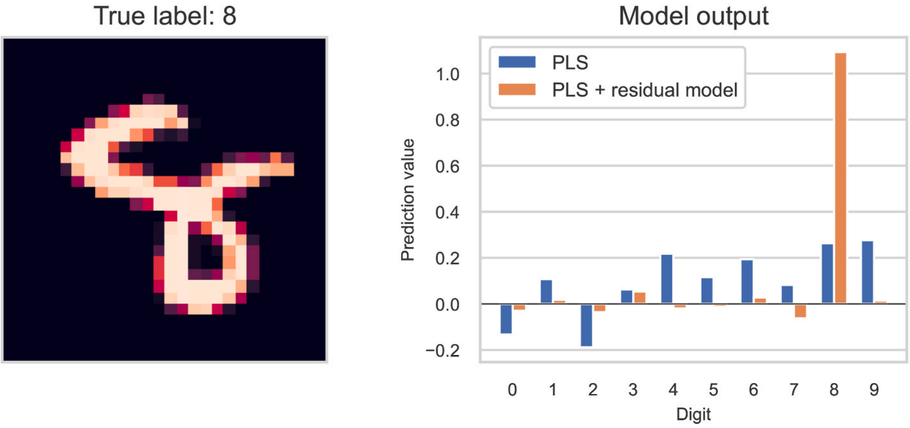


Fig. 13. Output from PLS model and after the MSE-shrinking approach for a sample image. True label: 8, PLS prediction: 9, prediction after residual shrinking: 8.

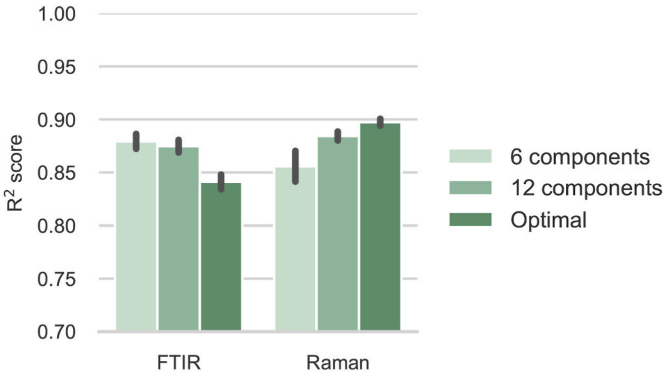


Fig. 14. R^2 score of the non-linear residual modelling using different numbers of PLS components. The error bars indicate the standard deviation of 10 different ANN weight initialisations.

predictions. The first PLS loading seems to be focused on capturing oval shapes like zeros consistent with the earlier interpretation from the score plot. The other loadings also resemble structures of familiar handwritten numbers.

The residual modelling approach offers additional options regarding model interpretation by studying how the final residuals are changed compared to the PLS residuals. For regression problems, the changes can be visualised as line plots as in Fig. 11. In this figure, the absolute values of the residuals of the test set predictions are shown. The blue line in the figure corresponds to the PLS model residuals plotted in increasing order from left to right. The corresponding new residuals after the residual shrinking are shown by the grey line. The red line corresponds to these new residuals after the residual shrinking in increasing order. It can be observed that the PLS residuals with large absolute values tend to get reduced by the ANN, while some samples get a higher residual after performing the residual modelling. The plot reveals the gain of the residual modelling, and may be useful for identifying possible outliers, e.g., samples with notably larger residuals compared to the overall residual distribution. The indicated outlying samples can then be inspected with respect to eventual irregularities in the data collection procedure.

In classification problems, a simple diagnostic tool to study the effect of the residual modelling are confusion matrices as shown in Fig. 12. A confusion matrix is used to evaluate a model performance by giving a summary of the counts of true versus predicted classes. By convention, the confusion matrix rows indicate the true class and the columns indicate the predicted class. For a useful classification model, the diagonal entries, showing the number of correctly classified samples, should contain the largest numbers. Here, these numbers are summarised as a bar plot for good overview. The confusion matrices reveal classes that are more challenging than other. For instance, the PLS model predicts the digit 7 to be a 9 wrongly 71 times. However, this mistake, along with most others, are corrected by the ANN model. Additionally, one can look at individual samples which the PLS model predicted wrongly and corrected by the residual shrinking, with the aim to pinpoint where the linear model works poorly. An example image is shown in Fig. 13 where the PLS model predicted the digit 9 instead of the correct digit 8. The barplot shows that the PLS model was not very confident in its decision among the alternative candidates (8, 4 and 6). Interestingly, the model

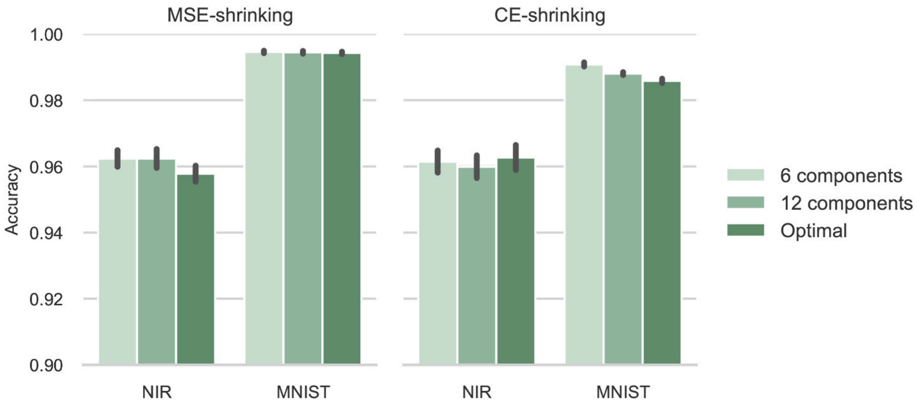


Fig. 15. Accuracy score of the non-linear residual modelling for classification using different numbers of PLS components. The error bars indicate the standard deviation of 10 different ANN weight initialisations.

obtained after the residual estimation is extremely confident in its prediction of the class to be an 8.

4. Discussion

The main benefit of the presented non-linear residual model scheme is that model interpretations are partly available, i.e., from the linear part of the modelling. In cases where the linear part of a hybrid model dominates the contribution to prediction it also provides better insights than the pure ANN modelling alternative. The examples shown in the results section indicate that non-linear ANN-modelling of the linear PLS residuals yields models that are often close to the pure ANN models in performance. The utility of the non-linear residual modelling becomes most evident when the problem is sufficiently complex and may serve as a useful alternative to a black-box ANN.

Our results also indicate that ANN based models are not always the superior choice regarding good predictions. For both the Raman- and NIR datasets analysed in this study, the difference in predictive performance between the ANN- and PLS models were insignificant. For such simple problems, it is therefore not surprising that the idea of hybrid modelling adds little or nothing with regard to improved predictive performance.

It is noteworthy that both the MSE-shrinkng and CE-shrinking approaches for classification performed well despite differences in the corresponding optimisation problems (i.e., different loss functions). No conclusion can be drawn whether one of the two approaches is superior based on the examples considered here. One major difference between the two alternatives is that the CE-shrinking requires tuning of an additional hyperparameter to set the balance between how much the PLS- and the ANN model contribute to the hybrid model predictions.

A fair question to ask is whether the proposed hybrid non-linear residual modelling is necessary or if one alternatively could train a PLS model to support interpretations and a separate ANN model to handle the predictions. However, this alternative provides no direct connection between the separate PLS- and ANN model behaviours. With the hybrid residual modelling scheme it will always be possible to inspect a prediction to assess its composition with regard to the linear and non-linear contributions. In effect, the proposed hybrid modelling can be thought of as shrinking the size of the ”black box” associated with the ANN model, where the shrinkage depends on the dominance of the linear model part.

Supported by prior knowledge regarding the data, subject to the model building, it is possible to judge the validity of a model by considering the content of the different model visualisation techniques. The diagnostic tools for interpretation presented can roughly be divided into two categories. On the one hand, the score- and loading plots provide insight into the subspace spanned by the PLS components. The other type of visualisation is provided by filters and feature maps of the neural network and governs a different feature space than the PLS model. The choice of visualisation from the ANN is dependent on the network architecture, and currently convolutional layers seem to offer the best options. Currently, it is easier to relate to visualisation of images as shown with the MNIST dataset. But, 1D signals can be treated in a similar fashion to detect regions of higher activation [7].

The residual modelling is not the only way of modelling non-linearities with a PLS model. Alternatively, a non-linear extensions of PLS modelling such as RBF-PLS [35] can be considered. This model might be a good alternative but lack the same flexibility in feature representations the ANN provides. Interpretability through the use of kernel functions might also be challenging.

Since the MNIST dataset consists of 2D images forming a three-dimensional tensor representation of input data, $\underline{\mathbf{X}}$, an alternative to vectorising (unfolding) the images to be features for ordinary PLS modelling is the application of a multilinear PLS version like N-PLS or N-CPLS [36,37] directly with the tensor representation. These methods also provide loading weights and loadings along each of the image dimensions which can be beneficial for understanding the resulting model. However, when comparing these modelling alternatives for the MNIST data, performance was near identical to ordinary PLS.

In our examples, non-linear residual modelling was applied on PLS models using the optimal number of components decided by cross-validation. It can be argued that using fewer components might be beneficial for the subsequent residual modelling approach. With fewer PLS components, the neural network essentially gets more freedom since the PLS model does not explain as much variance. Furthermore, since interpretation is typically done using the first few components, much of the insights are retained. Using fewer PLS components might also reduce the risk of letting the PLS model attempt to model non-linearities more suited for the ANN. To test the effect of using different numbers of components, all examples were repeated using 6 and 12 PLS components respectively, and keeping everything else unchanged, while repeating the ANNs on the new and larger residuals. A presentation of the results to be compared are shown in Figs. 14 and 15. In general, the use of fewer PLS components resulted in better performance of the resulting hybrid model. An exception is the Raman dataset where the use of fewer PLS components gave significantly worse performance. A possible explanation for this is that the ANN training process was unable to find any relevant information in the residuals. Hence, the ANN was not able to successfully model the larger residuals from a reduced PLS model (fewer components), resulting in an overall worse performance compared to an optimal PLS model. The failure of the residual modelling is shown by the higher variance in the associated error bar for the 6 component model alternative.

5. Conclusion

We have demonstrated a hybrid modelling framework where an artificial neural network (ANN) is used to predict residuals from a linear model. The concept is presented for regression problems and later extended for classification tasks. In the residual modelling scheme, the data modelling is split into a linear and non-linear part, allowing the majority of the data modelling to be done by a linear interpretable model while the ANN is used to boost the prediction performance. It is shown that with the proposed framework, it is possible to achieve almost the same predictive performance as a pure ANN modelling but with the added benefit that a larger proportion of the model can be interpreted, thereby shrinking the black box of the ANN.

CRediT authorship contribution statement

Runar Helin: Methodology, Software, Writing – original draft. **Ulf Indahl:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Oliver Tomic:** Methodology, Supervision, Writing – review & editing. **Kristian Hovde Liland:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.aca.2023.341147.

References

[1] Svante Wold, Harold Martens, Herman Wold, The multivariate calibration problem in chemistry solved by the PLS method, in: Matrix Pencils, Springer, 1983, pp. 286–293.

[2] S. Wold, A. Ruhe, H. Wold, W.J. Dunn, The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses, SIAM J. Sci. Stat. Comput. 5 (1984) 735–743.

[3] M. Mamouei, K. Budidha, N. Baishya, M. Qassem, P.A. Kyriacou, An empirical investigation of deviations from the beer–lambert law in optical estimation of lactate, Sci. Rep. 11 (1) (2021), 13734–13734.

[4] A. Kasper, Einarson, Andreas Baum, Terkel B. Olsen, Jan Larsen, Ibrahim Armagan, Paloma A. Santacoloma, Line K.H. Clemmensen, Predicting pectin performance strength using near-infrared spectroscopic data: a comparative evaluation of 1-d convolutional neural network, partial least squares, and ridge regression modeling, J. Chemometr. 36 (2) (2022).

[5] Runar Helin, Ulf Geir Indahl, Oliver Tomic, Kristian Hovde Liland, On the possible benefits of deep learning for spectral preprocessing, J. Chemometr. 36 (2) (2022).

[6] Puneet Mishra, Dário Passos, Multi-output 1-dimensional convolutional neural networks for simultaneous prediction of different traits of fruit based on near-infrared spectroscopy, Postharvest Biol. Technol. 183 (2022).

[7] Jacopo Acquarelli, Twan van Laarhoven, Gerretzen Jan, Thanh N. Tran, M. Lutgarde, C. Buydens, Elena Marchiori, Convolutional neural networks for vibrational spectroscopic data analysis, Anal. Chim. Acta 954 (2017) 22–31.

[8] Uladzislau Blazhko, Volha Shapaval, Vassili Kovalev, Achim Kohler, Comparison of augmentation and pre-processing for deep learning and chemometric classification of infrared spectra, Chemometr. Intell. Lab. Syst. 215 (2021).

[9] Xiaolei Zhang, Tao Lin, Jinfan Xu, Xuan Luo, Yibin Ying, Deepspectra: an end-to-end deep learning approach for quantitative spectral analysis, Anal. Chim. Acta 1058 (2019) 48–57.

[10] Salim Malek, Farid Melgani, Yakoub Bazi, One-dimensional convolutional neural networks for spectroscopic signal regression, J. Chemometr. 32 (2018), e2977.

[11] Jialin Dong, Mingjian Hong, Yi Xu, Xiangquan Zheng, A practical convolutional neural network model for discriminating Raman spectra of human and animal blood, J. Chemometr. 33 (2019), e3184.

[12] Chenhao Cui, Tom Fearn, Modern practical convolutional neural networks for multivariate regression: applications to NIR calibration, Chemometr. Intell. Lab. Syst. 182 (2018) 9–20.

[13] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, Been Kim, A benchmark for interpretability methods in deep neural networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019.

[14] Greger Andersson, Peter Kaufmann, Lars Renberg, Non-linear modelling with a coupled neural network - pls regression system, J. Chemometr. 10 (5–6) (1996) 605–614.

[15] Mehdi Khashei, Ali Zeinal Hamadani, Mehdi Bijari, A novel hybrid classification model of artificial neural networks and multiple linear regression models, Expert Syst. Appl. 39 (3) (2012) 2606–2620.

[16] Peigen Yu, Mei Yin Low, Weibiao Zhou, Development of a partial least squares-artificial neural network (pls-ann) hybrid model for the prediction of consumer liking scores of ready-to-drink green tea beverages, Food Res. Int. 103 (2018) 68–75.

[17] M.A. Hussain, M. Shafiur Rahman, C.W. Ng, Prediction of pores formation (porosity) in foods during drying: generic models by the use of hybrid neural network, J. Food Eng. 51 (3) (2002) 239–248.

[18] Suresh Dara, Priyanka Tumma, Feature extraction by using deep learning: a survey, in: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 1795–1801.

[19] Mohammed Brahimi, Kamel Boukhalfa, Abdelouahab Moussaoui, Deep learning for tomato diseases: classification and symptoms visualization, Appl. Artif. Intell. 31 (4) (2017) 299–315.

[20] G. Peter, Zhang. Time series forecasting using a hybrid arima and neural network model, Neurocomputing 50 (2003) 159–175.

[21] Phatchakorn Areekul, Tomonobu Senjyu, Hirofumi Toyama, Atsushi Yona, A hybrid arima and neural network model for short-term price forecasting in deregulated market, IEEE Trans. Power Syst. 25 (1) (2010) 524–530.

[22] Guoying Wang, Lili Zhuang, Lufeng Mo, Xiaomei Yi, Peng Wu, Xiaoping Wu, Bag: a linear-nonlinear hybrid time series prediction model for soil moisture, Agriculture 13 (2) (2023) 379.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (56) (2014) 1929–1958.

[24] David M. Haaland, Edward V. Thomas, Partial least-squares methods for spectral analyses. 1. relation to other quantitative calibration methods and the extraction of qualitative information, Anal. Chem. 60 (11) (1988) 1193–1202.

[25] Puneet Mishra, Dário Passos, Federico Marini, Junli Xu, Jose M. Amigo, Aoife A. Gowen, Jeroen J. Jansen, Alessandra Biancolillo, Jean Michel Roger, Douglas N. Rutledge, Alison Nordon, Deep learning for near-infrared spectral data modelling: hypes and benefits, TrAC, Trends Anal. Chem. 157 (2022).

[26] Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning. Springer Series in Statistics, Springer, New York, 2009.

[27] Dário Passos, Puneet Mishra, A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks, Chemometr. Intell. Lab. Syst. 223 (2022).

[28] Colin White, Willie Neiswanger, Yash Savani, Bananas: Bayesian Optimization with Neural Architectures for Neural Architecture Search, 2019 arXiv: 1910.11858 [cs.LG].

[29] Martín Abadi, Ashish Agarwal, Barham Paul, Eugene Brevdo, Zhifeng Chen, Citro Craig, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vanhoucke Vincent, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Wattenberg Martin, Wicke Martin, Yu Yuan, Xiaoqiang Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from, tensorflow.org.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[31] Kenneth Aase Kristoffersen, Kristian Hovde Liland, Ulrike Böcker, Sileshi Gizachew Wubshet, D. Lindberg, Svein Jarle Horn, Nils Kristian Afseth, FTIR-based hierarchical modeling for prediction of average molecular weights of protein hydrolysates, Talanta 205 (2019) 12.

[32] Kristian Hovde Liland, Achim Kohler, Nils Kristian Afseth, Model-based pre-processing in Raman spectroscopy of biological samples, J. Raman Spectrosc. 47 (6) (2016).

[33] M. Graña, M.A. Veganzons, B. Ayerdi, Salinas Valley hyperspectral image, Data retrieved from Grupo de Inteligencia Computacional, https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes, https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[35] B. Walczak, D.L. Massart, The radial basis functions — partial least squares approach as a flexible non-linear regression technique, Anal. Chim. Acta 331 (3) (1996) 177–185.

[36] Rasmus Bro, Multiway calibration. multilinear pls, J. Chemometr. 10 (1) (1996) 47–61.

[37] Kristian Hovde Liland, Ulf Geir Indahl, Joakim Skogholt, Puneet Mishra, The canonical partial least squares approach to analysing multiway datasets—N-CPLS, J. Chemometr. 36 (7) (2022).