PAPER • OPEN ACCESS

Machine learning based quantification of synchrotron radiation-induced x-ray fluorescence measurements—a case study

To cite this article: A Rakotondrajoa and M Radtke 2021 *Mach. Learn.: Sci. Technol.* **2** 025004

View the article online for updates and enhancements.

MACHINE
LEARNING
Science and Technology

PAPER

Machine learning based quantification of synchrotron radiation-induced x-ray fluorescence measurements—a case study

A Rakotondrajoa[1,2] and M Radtke[1]

1 BAM Federal Institute for Materials Research and Testing, Richard Willstaetter Str 11, Berlin 12489, Germany
2 Laboratoire de Physique Nucléaire et Physique de l'Environnement (LPNPE), University of Antananarivo, Antananarivo 101, Madagascar

E-mail: martin.radtke@bam.de

Abstract

In this work, we describe the use of artificial neural networks (ANNs) for the quantification of x-ray fluorescence measurements. The training data were generated using Monte Carlo simulation, which avoided the use of adapted reference materials. The extension of the available dataset by means of an ANN to generate additional data was demonstrated. Particular emphasis was put on the comparability of simulated and experimental data and how the influence of deviations can be reduced. The search for the optimal hyperparameter, manual and automatic, is also described. For the presented case, we were able to train a network with a mean absolute error of 0.1 weight percent for the synthetic data and 0.7 weight percent for a set of experimental data obtained with certified reference materials.

1. Introduction

Machine learning or artificial intelligence has made impressive progress in the last few years.

Spectacular successes have included AlphaGo's victory over the acting GO Champion and the creation of photorealistic images with general adversarial networks. By far not as spectacular but already present in our everyday life are applications in the area of sales, marketing and service, for example spam filters based on machine learning or recommender systems for product suggestions in online trade and suggestions on YouTube.

With these premises in mind, the authors have tried to explore the usefulness of deep neural networks for applications in x-ray fluorescence (XRF), especially for the direct calculation of elemental concentrations from spectra. The general task is multi-output regression with deep neural networks, which falls into the scope of supervised learning [1]. This differs in some essential aspects from the successful applications of deep learning mentioned above.

Although, as with all applications of machine learning, making a prediction is in the focus of attention, the requirements and objectives differ significantly. In advertising industry applications, the goal is to be better with the prediction than with the random selection of an advertisement. In scientific applications, the results must be as precise as possible and must be checked on known examples. This precondition is made worse by the fact that a wrong prediction in advertising generally does not have too serious consequences, whereas in the scientific field, a mistake, e.g. in the determination of a toxic ingredient, can be life-threatening.

The second significant difference is the retrieval of training data. While Facebook or Google, for example, have almost unlimited access to cheap, extensive data collections, their acquisition for scientific applications is often time-consuming and expensive.

In recent years, there have been several examples of artificial neural networks (ANNs) models to be applied to XRF measurements. Most of them did not deal with the direct quantification of the measurements from the spectrum. Ref [2] used principal component analysis (PCA) as a preprocessor and subsequently ANN to generate calibration curves for light elements (C, N, Na, Mg, and P) in soil samples. The applied

ANN consisted of two hidden layers and four neurons. Ref [3] combined the elemental concentrations obtained by total-reflection XRF with ANN to classify cancer patients and healthy individuals successfully. Ref. [4] demonstrated that the quantitative determination of parts per million (ppm) amounts of mixtures of Re, Os, Ir, and Pt in a polyethylene matrix, which could not be done successfully by conventional methods, was made possible by the application of neural network, with a root mean square error of a few ppm. Similar results have been shown as well in [5] using ANNs to resolve the overlapping peaks of lead and sulfur, reducing the uncertainty up to a factor of two.

Bos *et al* have shown in [6] the performance of ANNs for modeling the Cr-Ni-Fe system. More than a 100 steel samples with large variations in composition were measured, and ANNs were found to be robust and to perform generally better than other methods for calibrating over large ranges. Here too, the intensities of the characteristic fluorescence peaks rather than the entire spectrum were used as input values. Ref. [7] has successfully demonstrated that it is possible to determine the copper content in ores directly from the spectrum. The copper was present in relatively small concentrations of between 0.1% and 3%. Due to the speed of calculations, the authors predicted a potential use in continuous, real-time and online measurements.

Common to all the examples shown is that they have used actual experimental data as training data. The number of examples for training and testing was therefore naturally relatively small, ranging from 360 in [7] to 8 in [4].

In the following, the use of an ANN as a direct quantification method for XRF measurement is shown. The simultaneously direct learning from spectra to element concentrations for all present main elements is demonstrated. The training data were generated with a Monte Carlo (MC) simulation to enable a sufficient data corpus.

We wanted to see to what extent the progress in algorithms and computing technology makes preparatory steps unnecessary. This paper will not and does not claim to present the ultimate solution to the problem presented here but tries to share the experiences the authors have made.

2. The challenge

XRF is the emission of characteristic x-rays from a material that has been excited by bombardment with high-energy x-rays or gamma rays [8]. It is a fast, non-destructive analytical technique for determining element concentrations suitable for a wide range of applications ranging from archaeology [9], biology [10], geology [11], environmental science [12] to astronomy [13].

Common to all applications of XRF is that the measured signal, i.e. the intensity of the characteristic radiation, must be converted into element concentrations. The conventional methods of quantification in XRF spectrometry generally consist of the following steps:

- normalization of the data;
- identification of the existing elements;
- spectrum fitting;
- determination of the concentrations of the components with
  - fundamental parameters (FP)/MC simulation;
  - methods based on standard samples.

Even if there is dedicated software like pyMCA [14] or QXAS [15] available, this is sometimes a complicated task. The main disadvantage is that the determination of concentrations based on FP or MC calculations must be done in an iterative way, as the elements influence each other by so-called second-order effects, especially absorbing the generated fluorescence radiation and submitting additional radiation subsequently. For this reason, the whole process is rather time-consuming. As shown in figure 1, this leads as well to a complicated relationship between peak intensity and concentration. Quantification via standard materials is often prevented by the lack of the required reference materials. An overview of XRF and the quantification methods can be found in [8]. For this study, synchrotron radiation-induced XRF was used. Synchrotron radiation is advantageous compared to conventional x-ray sources because the polarization allows an effective reduction of the background, and the high flux allows small beam sizes and fast data acquisition. Additionally, the use of monochromators allows the selection of optimal excitation conditions.

Today, many applications of XRF require that the concentrations are known during or immediately after the measurement. This requirement is necessary because real-time monitoring of these concentrations is required, e.g. process monitoring in the industry, or because there is a large amount of data that otherwise must be quantified after the measurement. The latter is the case, e.g. at the BAMline [17], the beamline of
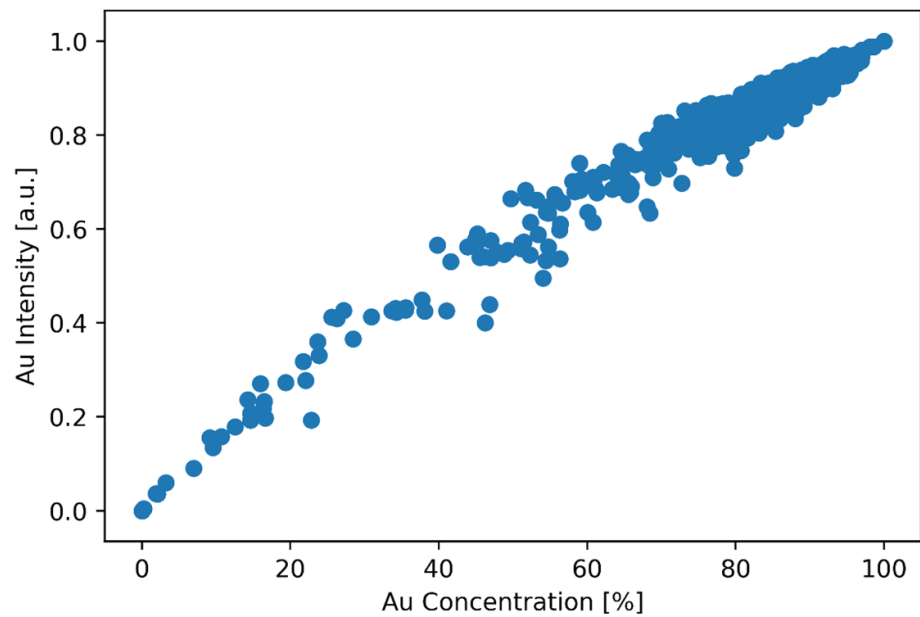
Figure 1. The calculated relation between measured intensity and the concentration of an element, in this example gold, is complicated. SNRLXRF [16] was used for this purpose and random concentrations of gold, silver, copper and tin were assumed for the data in this figure. It is clear, that the measured intensity is not only dependent on the concentration of the element itself, but as well on the concentrations of all other elements present. Therefore, matrix-matched reference materials or iterative calculations are needed for quantification.

BAM at BESSY, where a so-called color x-ray camera [18] is used as a detector for XRF measurements. This detector is based on a pn-CCD chip and produces 70 000 spectra simultaneously for one measurement.

For this device, it is practically impossible to quantify the spectra with conventional methods afterward, because the measurement time is significantly shorter than the amount of time required for the calculations. Instead, techniques that are capable of extracting concentrations directly from the spectra must be used. In this paper we used three measurements with a single conventional detector setup at BAMline for testing purposes, because this detector is better characterized, which is advantageous for the MC simulations.

According to the universal approximation theorem, a large enough deep neural network can approximate any function [19]. The task presented here is a multi-output regression problem. The input data are 950 channels of a spectrum normalized to a sum of 1, and the output layer is formed by the element concentrations in weight percent for the five elements present in the sample. As we know input and output data, we deal with a supervised learning problem. We expect that the network learns two tasks: first, the extraction of the intensities from the spectrum and afterward calculate concentrations with corrections for higher-order effects.

Parameters describing the network and the learning algorithm are called hyperparameters. Hyperparameters must be defined before the training and can be optimized to obtain the best results. For the first attempt, a neural network with three hidden layers and arbitrarily 50 neurons per layer was chosen. To overcome the vanishing gradient problem, rectified linear units (RELU) were selected as activation function [20] and state-of-the-art Adam optimizer [21] was used. Knowing that elemental concentrations are positive, RELU was selected as well for the output layer to inhibit negative values. The training was performed with 500 epochs, early stopping with a patience value of 50 was applied to avoid overfitting. As a loss function, the mean absolute error (MAE) was chosen. The loss function was calculated for each epoch, and the current model was saved when the value of the loss function for validation data reached a minimum.

All networks used in the following were created with different versions of Tensorflow [22]. In the process of this work, there was a major update, so the first experiments were done with Tensorflow 1.x and the last experiments with Tensorflow 2.2. For defining the network with TF2, the Keras library was used. If not explicitly mentioned, default values were applied.

Following the advice from [23], 'A dumb algorithm with lots and lots of data beats a clever one with modest amounts of it.' we did not try to exploit advanced network structures, like it has been done for x-ray photoelectron spectroscopy in [24], but instead tried to optimize the availability and quality of training data.

To keep the parameter space manageable, we have also constructed the networks in such a way that the neurons are distributed evenly over the hidden layer.

Table 1. Certified values for the gold reference materials.

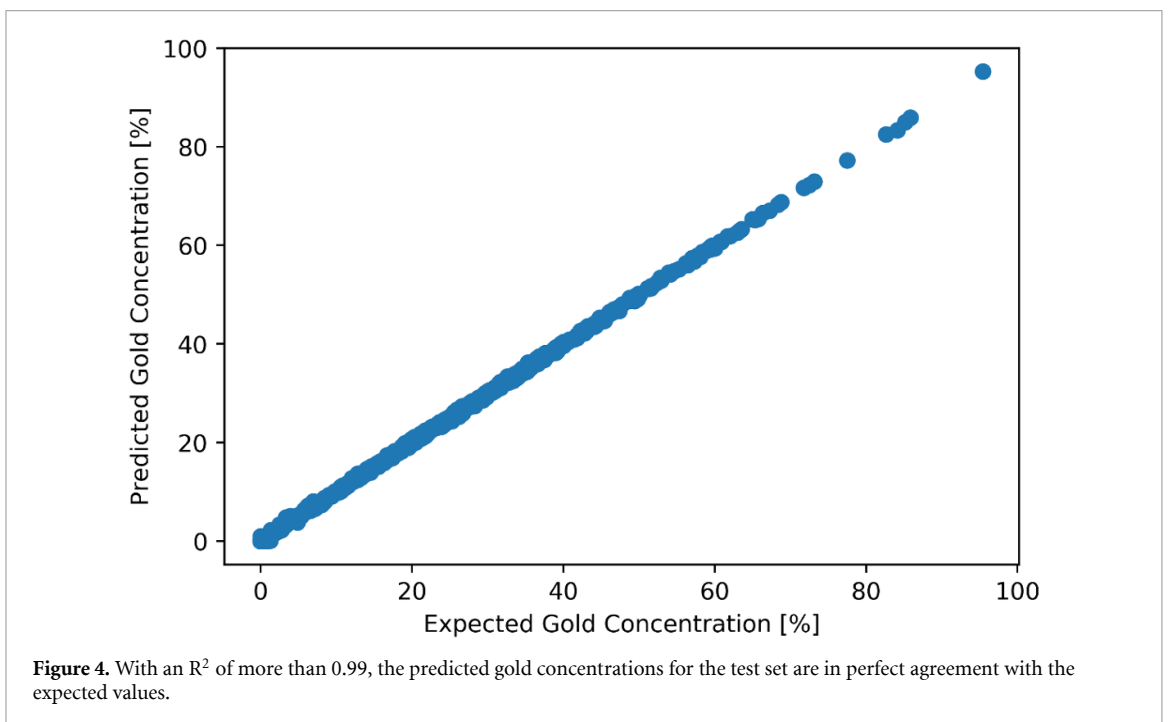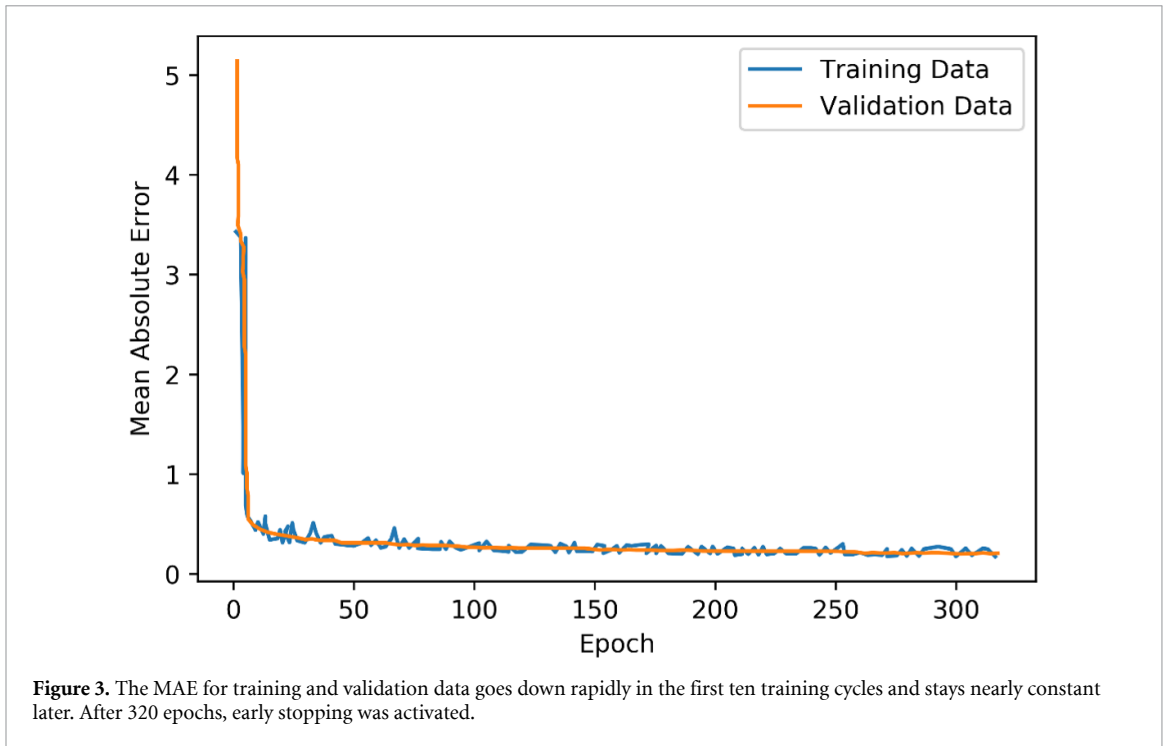|	Rose gold (ERM 506)	White gold (ERM 507)	Yellow gold (ERM 508)
Element	Mass fraction in weight %	Mass fraction in weight %	Mass fraction in weight %
Au	58.56 + 0.06	75.10 + 0.11	75.12 + 0.11
Ag	3.90 + 0.05	3.02 + 0.05	24.90 + 0.05
Cu	35.65 + 0.06	14.69 + 0.05	
Ni		4.99 + 0.04	
Zn	1.891 + 0.018	2.107 + 0.016	
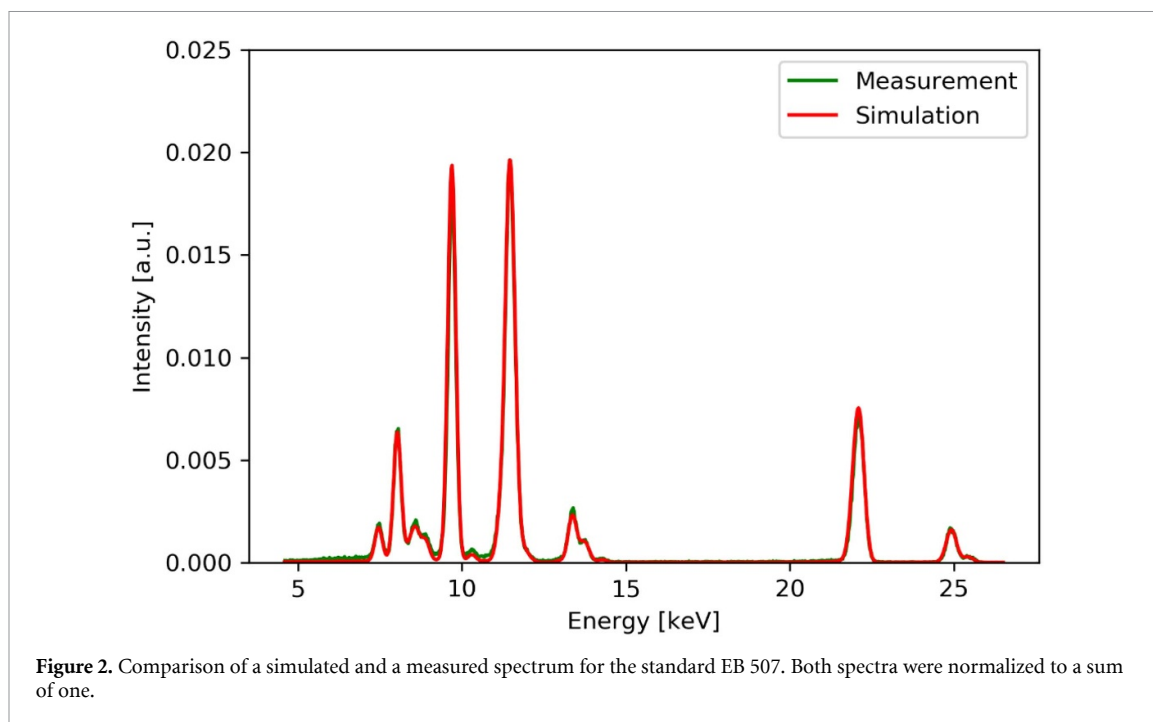
Figure 2. Comparison of a simulated and a measured spectrum for the standard EB 507. Both spectra were normalized to a sum of one.

3. Training data

As with all methods based on multilayer neural networks, the availability of a large amount of high-quality training data is a crucial point for success. As mentioned before, the availability of reference materials is an issue. Therefore, MC simulated data served as input. To generate the needed dataset, the program XMIMSIM [25] was used. With this program, it is possible to generate synthetic XRF spectra taking into account sample composition, measurement geometry, excitation conditions, and detector characteristics. All of these parameters have an influence on the measured spectra. The results of the simulations are virtually noiseless due to the high number of simulated photons.

Due to the high dimensionality of the full dataset, some restrictions were applied. All parameters were fixed, except sample composition. The sample composition was restricted to the five elements of interest: gold, silver, copper, nickel, and zinc. These elements are present in three BAM certified reference materials (CRMs: ERM-EB506, ERM-EB507, and ERM-EB508), which served as control samples to evaluate the results for real measurements. Elemental concentrations are given in table 1. Measurements from these samples were used as well to determine the needed parameters for the MC simulation. An example spectrum is shown in figure 2. With this configuration, 24 500 spectra were generated. For the calculations, the data was split into three parts: 80% of the data was used for training, 10% as validation data, and 10% as test data. Each of these parts has different tasks: the training dataset was used to train the weights and biases of the network, the validation dataset was used to calculate the performance while training, and, finally, the test dataset was used to obtain the final performance characteristics. The validation dataset as well was used to apply early stopping, which means that the training was interrupted if no gain in performance after a defined number of tries was reached.

Figure 3. The MAE for training and validation data goes down rapidly in the first ten training cycles and stays nearly constant later. After 320 epochs, early stopping was activated.

Figure 4. With an R^2 of more than 0.99, the predicted gold concentrations for the test set are in perfect agreement with the expected values.
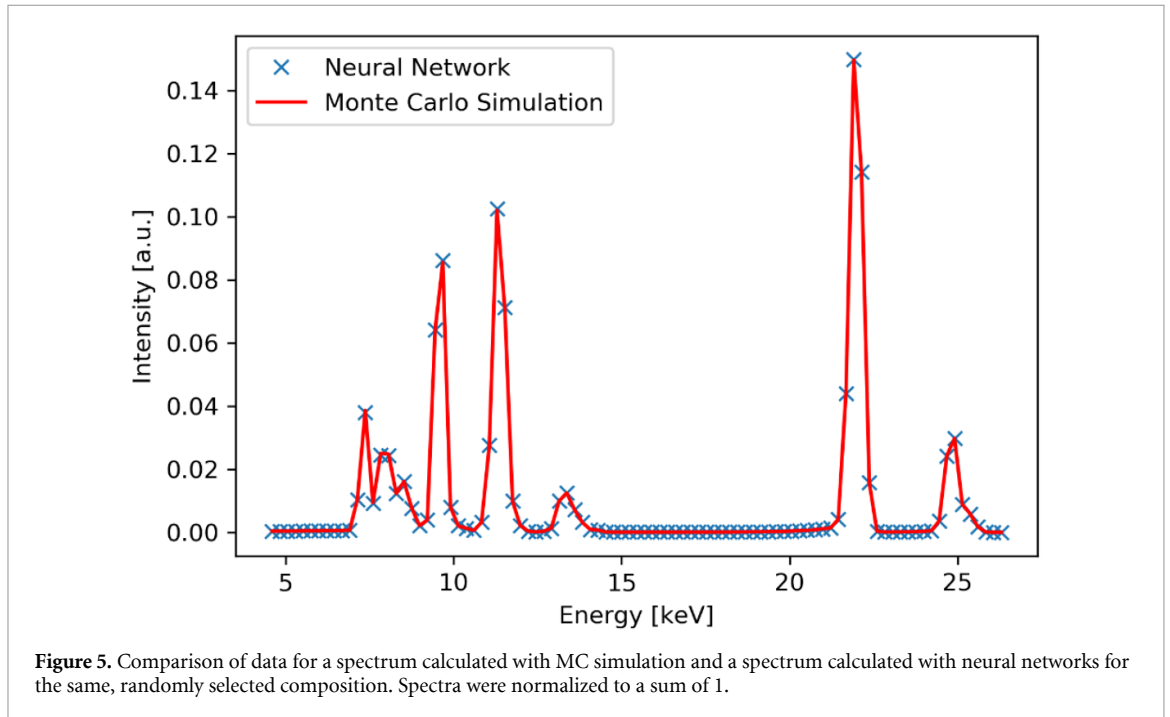
4. First results

Using the network and hyperparameters described in the previous paragraph, the network was trained on the simulated data. Figure 3 shows the model loss as a function of training epochs. It can be seen that the training was stopped after approximately 300 epochs because no more progress was made. No overfitting appears, as validation and training data have nearly the same MAE of 0.2. This means that we can predict concentrations in the validation set with an MAE of absolute 0.2%; for the test set, an MAE of 0.2 was achieved as well. Figure 4 shows expected and predicted values for gold for the test set, which have a perfect linear relation.

Anyhow, it has to be taken into account that up to now, only simulated data have been analyzed. For the gold CRMs, the results shown in table 2 were obtained. The MAE $= 1$ for the experimental data is around five times higher than for the simulated data. A clear trend can be seen here. At high concentrations, which correspond to well-defined peaks in the spectrum, the values obtained match the expected values reasonably.

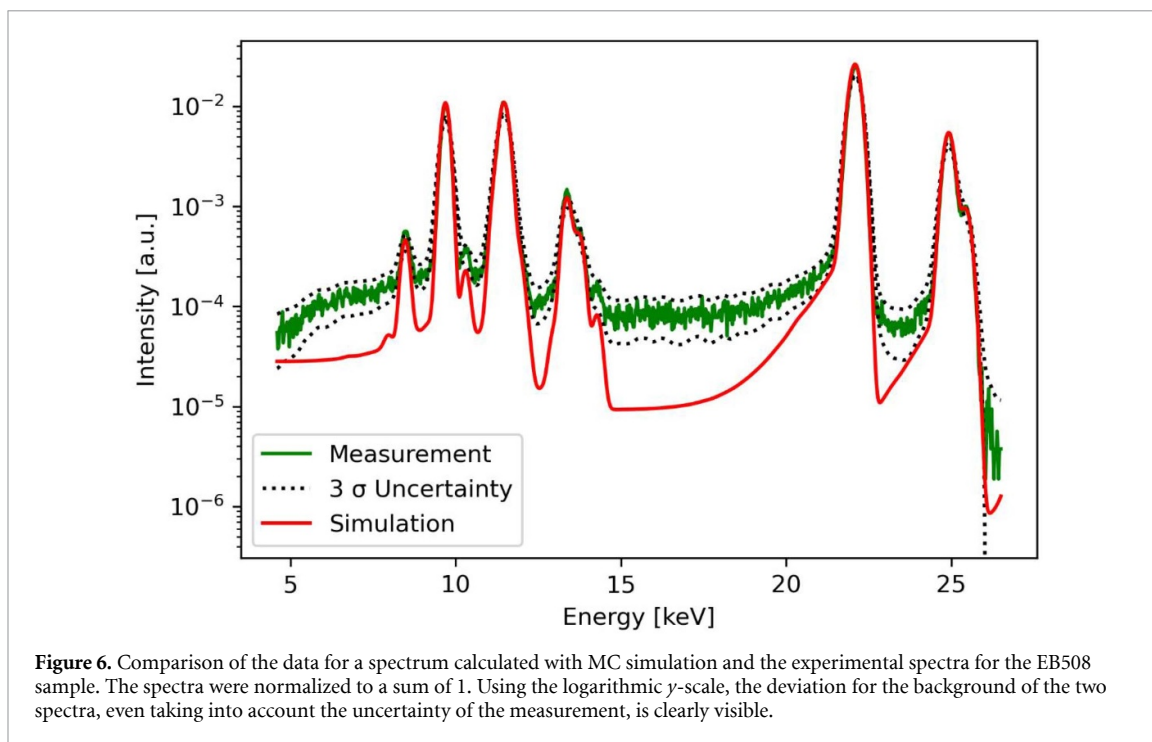**Table 2.** Predicted values for the gold reference materials in weight %.

	Ni	Cu	Zn	Ag	Au	Total
ERM 508	0.00	0.00	0.00	24.90	75.12	100.02
Predicted	1.95	2.20	0.00	22.75	72.55	99.45
ERM 506	0.00	35.65	1.89	3.90	58.56	100.00
Predicted	0.00	35.71	1.75	4.19	57.47	99.13
ERM 507	4.99	14.69	2.11	3.02	75.10	99.91
Predicted	5.46	15.47	2.31	3.40	73.00	99.64

Figure 5. Comparison of data for a spectrum calculated with MC simulation and a spectrum calculated with neural networks for the same, randomly selected composition. Spectra were normalized to a sum of 1.

Problems arise in particular with the elements that are not expected to be in the sample. Another problem which becomes visible here as well is that only concentrations are given. There is no assessment for the uncertainty of the predicted values.

5. Approaches for optimization

There are principally two ways to enhance the results obtained so far: The training data can be improved, or the network structure can be optimized. As mentioned above, the literature suggests that a larger amount of training data has a positive influence on the results. The difficulty here is that the MC simulations are very time-consuming. However, a large amount of data is already available, so that a different approach can be taken. By using the concentrations as input and the spectra as output values, a neural network has been trained to predict the spectra depending on the concentration. In order not to make the size of the dataset to be processed too large, the 950 spectral channels were rebinned into 95 channels. To predict the spectra, a deep neural network with 5 input channels, corresponding to five element concentrations, and 95 output channels was created. One hidden layer with 50 neurons and one hidden layer with 500 neurons were used. Figure 5 shows that the data generated by simulation and the data generated with the neural network match well. Therefore no further attempts to optimize this network were made. In this way, within a few hours, a network has been trained, which was then able to generate 1 000 000 training datasets within a few minutes. The idea of augmenting the training data with synthetic data from a neural network trained with a smaller dataset, calculated with MC simulation, was demonstrated very successfully by Carminati *et al* in [26] by generating training data for high-energy physics detectors. While the application shown here is much more modest, the distinct advantages remain the same. First, because of the faster data generation, a more fine-meshed dataset can be used for training. Second, since the generative network averages over all examples, the simulation can be performed with worse statistics, i.e. in less time without compromising the precision of the final results. This advantage was not used in this work, because the simulations were already performed with high precision, but this fact can be used advantageously in future projects.

Figure 6. Comparison of the data for a spectrum calculated with MC simulation and the experimental spectra for the EB508 sample. The spectra were normalized to a sum of 1. Using the logarithmic *y*-scale, the deviation for the background of the two spectra, even taking into account the uncertainty of the measurement, is clearly visible.

However, the main idea of this work is to quantify the XRF spectra directly. Therefore the additional step of rebining is not desired but is due to the restricted computational resources

Apart from the amount of training data, the quality is, of course, also important. If one considers the results shown in table 2 and the comparison of the measured and simulated spectrum in figure 6, it becomes immediately apparent that the problem for small concentrations originates in the non-optimal agreement of the background in simulated and measured spectra.

In the following, an attempt was made to solve this problem by adding to each training spectrum an additional background modeled by a polynomial generated with random parameters. It was expected that the neural network learns this background and corrects the extractions of the intensities, and therefore the concentrations, accordingly, especially for small peaks.

The second approach is to improve the network structure and the hyperparameters. For the network structure, this is the number of layers and the number of neurons per layer. For the hyper-parameters, this is essentially the number of epochs, the optimizer used, the learning rate, mini-batch size, and the activation function. Again, the problem is that the parameter space, and thus the number of computations required, quickly becomes too large. For this work, we have therefore limited ourselves to the number of neurons per layer.

In total, six different scenarios were tested: the original dataset, the original dataset rebined to 95 channels, and the dataset generated from the rebined data with the neural network. Each of these datasets was available once in its original form and once with an artificially generated background.

All these datasets were used to train four networks, with 25, 50, 100, or 200 neurons per layer, so that a total of 24 different networks were trained. The maximum number of training epochs was set to 2000 with early stopping and a patience value of 200.

A summary of the results obtained is given in table 3 in the next section.

Although it is clear from the obtained values, that with an MAE below one, results that are already usable have been obtained with this procedure, it is also immediately apparent that the selection of the hyperparameters was relatively arbitrary. To overcome this, in a final optimization step, the so-called autoML or automated machine learning was tested. autoML is the name for a group of autonomous hyperparameter optimization tools that became popular recently and optimize the hyperparameter autonomously. This can be realized, e.g. by grid or random search. We decided to use another approach, the so-called Bayesian optimization combined with Gaussian processes [27]. This method is advantageous when dealing with a large parameter space and high costs, in our case long computing times, for the evaluation. In simple words, it tries to minimize the number of search steps required by adapting a model, including the uncertainty, to the underlying parameters. We have used the Python implementation from [28]. Because of the complete documentation and the easy applicability, we preferred it over the built-in autoML tool in Keras.

Table 3. The results for the MAE, obtained with a simple grid search. Best results for each datasource are marked in bold.

Datasource	Background	N-neuron	MAE MC	MAE CRM
MC	N	25	4.4	1.5
MC	Y	25	1.3	2.1
MC	N	50	4.3	3.4
MC	Y	50	1.3	1.7
MC	N	100	4.3	3.1
MC	Y	100	0.8	**0.7**
MC	N	200	**0.3**	1.4
MC	Y	200	1.2	1.0
MC/Rebin	N	25	0.3	**0.7**
MC/Rebin	Y	25	1.3	1.1
MC/Rebin	N	50	**0.1**	0.8
MC/Rebin	Y	50	1.1	2.0
MC/Rebin	N	100	**0.1**	**0.7**
MC/Rebin	Y	100	1.0	1.1
MC/Rebin	N	200	**0.1**	1.0
MC/Rebin	Y	200	0.6	1.0
NN	N	25	0.5	2.7
NN	Y	25	1.1	1.5
NN	N	50	0.2	3.9
NN	Y	50	2.1	1.5
NN	N	100	**0.1**	1.3
NN	Y	100	1.3	**0.7**
NN	N	200	**0.1**	1.9
NN	Y	200	0.5	0.5

On the other hand, one has to consider that certain parameters have to be specified for the autoML algorithm as well. First of all, it has to be decided which parameters should be optimized, in our case, the number of layers, the number of neurons per layer, the number of examples for batch analysis, and the learning rate. Of course, reasonable boundaries, within which these parameters can be varied, must also be defined.

Moreover, the optimization algorithm itself also requires the specification of some other parameters. These parameters determine how the search behaves. For example, one has to decide whether the algorithm is more explorative or exploitative and whether it should be more interested in minimizing the overall uncertainty or the efficiency of the search. Therefore, it must be clear that with autoML, in a sense, one is just trading one problem for another. We decided to optimize the number of layers, the number of neurons per layer, batch size, and learning rate.

Since this procedure allows much more efficient testing, we have done another manipulation of the input data here. The addition of an artificial background causes a shift in the y-axis. In order to simulate small fluctuations of the measurement setup, a small random offset for the x-axis was additionally used in the training data. At last, a Gaussian filter with randomly selected width between 0 and 3 was applied. This should force the network to learn small instabilities in the measurements, which can influence peak positions and width. The results obtained are listed in table 4 in the following section.

6. Results and discussion

The results of the manual optimization are shown in table 3. Basically, a very rough grid search was performed by the number of neurons per layer, addition of background, and data source.

In general, one can see here that the accuracy of the model increases with an increasing number of neurons. As expected, the accuracy of the test dataset from MC simulated data (MAE MC) is better than for the measured values (MAE CRM). Better MAE values are achieved for the rebined (MC/Rebin) and neural network generated data (NN) than with the training dataset (MC). Furthermore, it can be seen that, in the majority of cases, the MAE MC values are smaller when no background is added, and the MAE CRM values are smaller when a background is added. This corresponds precisely to the expected and desired behavior. We can conclude from this that the networks learn to subtract the background successfully.

However, as mentioned above, this manual search covers only a fraction of the parameter space. Therefore, the next step was Bayesian optimization.

Table 4 shows the results of seven different conditions. The search space was extended to four parameters. For the first four conditions, the number of layers was limited between 2 and 4, the number of neurons per

Table 4. The results for the MAE, obtained with Bayesian optimization. The different conditions are explained in the text. Best results for 950 channel input data are marked in bold.

Condition	Batch size	Learning rate	N-layer	N-neuron	MAE MC	MAE CRM
1	1	0.009	2	168	**0.1**	1.3
2	1	0.009	2	343	0.8	**0.8**
3	1	0.009	3	381	1.3	**0.8**
4	1	0.009	2	343	1.2	1.0
5	2	0.002	4	352	0.9	1.4
6	1	0.002	5	19	1.8	0.9
7	1	0.003	3	377	0.7	0.7

layer between 25 and 500, the minibatch size between 1 and 64, and finally, the learning rate between 10–1 and 10–5. As acquisition function 'Expected Improvement' with the focus on exploration was chosen [28].

For condition 5, the acquisition function "Probability of Improvement" was selected for comparison. In condition 6, the maximum number of neurons per layer was restricted to 25, but the number of allowed layers was increased to 10.

The input data were gradually adapted to resemble the experimental data more closely.

In condition 1, the original MC spectra were used. In condition 2, an artificial background was added. To account for instabilities during the measurements in condition 3, a random shift of ± 2 channels in the x-axis was added, and in condition 4 additionally the training spectra were smoothed with a Gaussian filter with a random width between 0 and 3. All the spectra modifications have been done in the training and validation data, but not in the final test data.

Condition 5 is equal to condition 4, except for the use of the different acquisition function. Condition 6 is equal to condition 4, except that the network structure was forced to use fewer neurons per layer.

In condition 7, spectra generated with the neural network and added background were used.

For each condition, five random networks and 15 with optimized hyperparameters were trained.

From table 4, it can be seen that the simulated data were clearly best adjusted in the non-manipulated case 1. The more the input data were adapted to the real data, the larger the MAE MC becomes. However, it can also be observed here that the MAE CRM becomes smaller, the adaptation for the measured data, and thus the generalization of the network, becomes better. Therefore, this approach to overcome differences between simulated and measured data was successful. Interestingly, in condition 7, the use of the neural network generated data yields as well very good, for the MAE CRM even the best, results. Compared to the manual grid search, no outstanding improvement has been reached using Bayesian optimization. Anyhow, the authors feel more confident that the reached solution is near to optimal.

In conclusion, a few remarks on the practical implementation of similar projects should be made here.

First of all, experience has shown that the quantity and quality of training data are the most critical factors for success. As in scientific questions, the cost of these are generally high, the use of synthetic data produced with simulations or neural networks is beneficial. Due to the wide range of the possible experimental parameter, it seems unrealistic to train a network only with experimental data. For this purpose, it must be ensured that the experimental and simulated data match as closely as possible. On the bright side, as shown, the adaption of experimental and simulated data can be made with relatively crude measures.

It is noticeable that the optimized networks are all relatively large compared to old works. This is probably due to the use of early stopping. Since this reliably avoids overfitting, there is no incentive for the optimizer to use small networks. For the networks used here, this is still tolerable, but for more extensive networks, one should think about implementing a penalty in the cost function for large networks.

Once the network has been trained, the evaluation per spectrum is very fast, typically about 20 ms. However, the time required to train the network, typically one hour for one set of hyperparameters, and especially to obtain the training data is significant. It is therefore essential to make sure that the effort is worth it, i.e. that the trained network can be used for many cases,

The final remark is that for successful completion of the project, the goal, in this case the uncertainty or the maximum MAE which needs to be achieved, should be defined in advance. Machine learning is such a broad and dynamically developing field that there is literally no limit for optimizing. Here we have restricted ourselves to a basic ANN architecture and the dispensing of additional spectra manipulation like PCA. Even so, the number of possible variations is immense. In total, we have trained more than 500 networks for this work. Therefore, autoML, which can take over the part of network optimization, is a major advance, and we believe it will gain significant importance in the future.

.

7. Conclusion

We have shown that it is possible to quantify the concentrations of elements from XRF measurements with neural networks directly from spectra. As an example, gold samples with nickel, copper, zinc, and silver contents were used. All main elements were determined simultaneously; reference materials certified by BAM served as control samples.

Synthetic spectra generated with MC simulations were used as training data. Due to the relatively easy availability of these data, it was possible to apply significantly larger networks than in previous work. The enhancement of the available dataset using an ANN to generate additional data was demonstrated. It has been shown that this approach provides reliable results over the full concentration range from 0% to 100%. An uncertainty of less than one percent absolute MAE was achieved. The most significant difficulty was to ensure the agreement of experimental and simulated data. This could be obtained by adding additional backgrounds and shifting the spectra on the energy axis. A challenge for the practical application is that the training data depends on the sample composition, the excitation conditions, the geometry and the detection system. Therefore, the data and the trained network cannot simply be reused for other conditions. In general, however, it must be said that the relatively long time required for training and model generation can only be justified if a high degree of reusability of the model is ensured. This is mainly to be expected with commercial instruments which are used under constant measurement conditions for a constant type of sample. In this case, real-time quantification can be achieved.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

ORCID iD

M Radtke ● https://orcid.org/0000-0003-0014-8905

References

[1] Borchani H, Varando G, Bielza C and Larrañaga P 2015 A survey on multi-output regression *Wiley Interdiscip. Rev. Data Min. Knowl. Discovery* **5** 216–33
[2] Kaniu M I, Angeyo K H, Mwala A K and Mangala M J 2012 Direct rapid analysis of trace bioavailable soil macronutrients by chemometrics-assisted energy dispersive X-ray fluorescence and scattering spectrometry *Anal. Chim. Acta* **729** 21–25
[3] Hernandez-Caraballo E A and Marco-Parra L M 2003 Direct analysis of blood serum by total reflection x-ray fluorescence spectrometry and application of an artificial neural network approach for cancer diagnosis *Spectrochim. Acta* B **58** 2205–13
[4] Boger Z and Karpas Z E 1994 Application of neural networks for interpretation of ion mobility and x-ray fluorescence spectra *Anal. Chim. Acta* **292** 243–51
[5] Facchin I, Mello C, Bueno M I M S and Poppi R J 1999 Simultaneous determination of lead and sulfur by energy-dispersive x-ray spectrometry. Comparison between artificial neural networks and other multivariate calibration methods *X-Ray Spectrom.* **28** 173–7
[6] Bos A, Bos M and van der Linden W E 1993 Artificial neural networks as a multivariate calibration tool: modeling the Fe-Cr-Ni system in x-ray fluorescence spectroscopy *Anal. Chim. Acta* **277** 289–95
[7] Escarate P, Hein R, Durán M and Ramaciotti P 2015 X-ray fluorescence spectroscopy for accurate copper estimation *Miner. Eng.* **71** 13–15
[8] Beckhoff B *et al* 2007 *Handbook of Practical X-Ray Fluorescence Analysis* (Heidelberg: Springer) 9783540367222
[9] Radtke M, Reinholz U and Gebhard R 2017 Synchrotron radiation-induced x-ray fluorescence (SRXRF) analyses of the Bernstorf gold *Archaeometry* **59** 891–9
[10] Hachmoller O *et al* 2016 Elemental bioimaging and speciation analysis for the investigation of Wilson's disease using μXRF and XANES *Metallomics* **8** 648–53
[11] Buhn B, Rankin A H, Radtke M, Haller M and Knoechel A 1999 Burbankite, a (Sr,REE,Na,Ca)-carbonate in fluid inclusions from carbonatite-derived fluids: identification and characterization using laser Raman spectroscopy, SEM-EDX, and synchrotron micro-XRF analysis *Am. Mineral.* **84** 1117–25
[12] Strub E *et al* 2008 Determination of Cr(VI) in wood specimen: a XANES study at the Cr K edge *Nucl. Instrum. Methods Phys. Res.* B **266** 2405–7
[13] Fabian A C, Rees M J, Stella L and White N E 1989 X-ray-fluorescence from the inner disk in Cygnus X-1 *Mon. Not. R. Astron. Soc.* **238** 729–36
[14] Sole V A, Papillon E, Cotte M, Walter P and Susini J 2007 A multiplatform code for the analysis of energy-dispersive x-ray fluorescence spectra *Spectrochim. Acta* B **62** 63–68

[15] He F and Vanespen P J 1991 General-approach for quantitative energy dispersive-x-ray fluorescence analysis based on fundamental parameters *Anal. Chem.* **63** 2237–44

[16] Keenan J A and Holmes D 1982 Quantitative trace analysis by x-ray-fluorescence using the computer-program NRLXRF *Appl. Spectrosc.* **36** 19–22

[17] Riesemeier H, Ecker K, Görner W, Müller B R, Radtke M and Krumrey M 2005 Layout and first XRF applications of the BAMline at BESSY II *X-Ray Spectrom.* **34** 160–3

[18] Scharf O *et al* 2011 Compact pnCCD-based x-ray camera with high spatial and energy resolution: a color x-ray camera *Anal. Chem.* **83** 2532–8

[19] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press) 978-0262035613

[20] Dahl G E *et al* 2013 Improving deep neural networks for LVCSR using rectified linear units and dropout *2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (New York: IEEE) pp 8609–13

[21] Ruder S 2016 An overview of gradient descent optimization algorithms (arXiv:1609.04747)

[22] Abadi M *et al* 2016 Tensorflow: a system for large-scale machine learning *12th USENIX Symposium on Operating Systems Design and Implementation ((OSDI) 16)* (arXiv:1605.08695)

[23] Domingos P 2012 A few useful things to know about machine learning *Commun. ACM* **55** 78–87

[24] Drera G, Kropf C M and Sangaletti L 2020 Deep neural network for x-ray photoelectron spectroscopy data analysis *Mach. Learn.: Sci. Technol.* **1** 015008

[25] Schoonjans T, Vincze L, Solé V A, Sanchez Del Rio M, Brondeel P, Silversmit G, Appel K and Ferrero C 2012 A general Monte Carlo simulation of energy dispersive x-ray fluorescence spectrometers—part 5 polarized radiation, stratified samples, cascade effects, M-lines *Spectrochim. Acta* B **70** 10–23

[26] Carminati F *et al* 2018 Three dimensional generative adversarial networks for fast simulation *J. Phys.: Conf. Ser.* **1085** 032016

[27] Frazier P 2018 A tutorial on Bayesian optimization (arXiv:1807.028112)

[28] Nogueira F 2014 Bayesian optimization: open source constrained global optimization tool for python (available at: https://github.com/fmfn/BayesianOptimization)