

Determination of ^{239}Pu content in spent fuel with the SINRD technique by using artificial and natural neural networks

Borella Alessandro¹, Riccardo Rossa¹, Hugo Zaioun^{1,2}

¹SCK•CEN, Boeretang 200, B2400 Mol, Belgium

²IMT Atlantique, Campus de Nantes, rue Alfred Kastler 4, 44307 Nantes, France

Email: aborella@sckcen.be

Abstract:

In the last years, a database of simulated spent fuel observables was developed at SCK•CEN by combining the results of depletion-evolution codes and the responses of several detectors obtained by Monte Carlo models. We analysed the large amount of generated data with Artificial Neural Networks, by using the MATLAB toolbox.

In this paper we focus on the application of Artificial Neural Networks to simulated Self-Interrogation Neutron Resonance Densitometry observables with the aim to quantify the ^{239}Pu content in spent fuel. In view of a realistic application of the method, the number of data in the training and validation sets was limited to 20 spent fuel assemblies; the obtained performance when using randomly selected spent fuel assembly was compared with the one obtained when the spent fuel assemblies were selected by expert judgement. The average deviation between the nominal ^{239}Pu content and the calculated ^{239}Pu content in the testing data set was 0.2% with a standard deviation of 3.5% and a maximum deviation of 10%.

It was found that the selection of spent fuel assemblies based on expert judgement results in better performances and therefore speeds up the data analysis when compared to a pure random selection of the data; hence the term natural, as opposite to artificial, is present in the title of the paper.

Keywords: Spent Fuel; Non Destructive Analysis observables; Artificial neural networks; Self-Interrogation Neutron Resonance Densitometry; Large Data sets

1. Introduction

Non-destructive assay (NDA) of spent fuel assemblies (SFA), either for safeguards verification purposes or for safety aspects related to nuclear fuel cycle, relies often on the detection of neutron and gamma radiation spontaneously emitted by the spent fuel [1]. Gamma radiation is mainly emitted by fission products and therefore its measurement does not represent a direct assay of the quantity of fissile material present in the SFA. Neutron radiation is originating mainly from spontaneous fission decay and α -decay via (α ,n) reactions on oxygen isotopes. The decay

of actinides such as Cm isotopes represents the main source of such neutron radiation which can then undergo subsequent multiplication in the fissile material of the fuel. Therefore, also the observables associated to neutron measurements on SFA do not represent a direct assay of the quantity of fissile material present in the SFA, unless one is able to determine the multiplication and relate that to the residual fissile mass [2].

In this framework, and in relation to increased verification needs associated to the imminent start of operation of geological repositories [3,4], R&D on NDA intensified in the last decade [5,6,7,8,9]. One of the techniques that was studied is the Self-Interrogation Neutron Resonance Densitometry (SINRD) [10,11]. The observables associated to this technique are directly related to the quantity of ^{239}Pu in the fuel and therefore have the potential to provide means for a direct quantification of the ^{239}Pu amount in a SFA.

The use of the SINRD technique and the data analysis of the associated observables by means of artificial neural network (ANN) [12,13] is described in the paper.

2. Self-Interrogation Neutron Resonance Densitometry observables

The SINRD technique [10] is a NDA technique for the direct quantification of ^{239}Pu . The total neutron cross-section of ^{239}Pu shows a strong resonance around 0.3 eV and the attenuation of the neutron flux around the 0.3 eV energy region is used to directly quantify the ^{239}Pu mass. The passive neutron emission from spent fuel is measured with fission chambers bare or wrapped with different absorbers as follows:

- Cd wrapped ^{235}U fission chamber, insensitive to thermal neutrons
- Bare ^{235}U fission chamber, mainly sensitive to thermal neutrons
- Bare ^{238}U fission chamber, mainly sensitive to fast neutrons
- ^{239}Pu fission chamber covered by Gd foil, sensitive to neutrons with energy > 0.1 eV
- ^{239}Pu fission chamber covered by Cd foil, sensitive to neutrons with energy > 1 eV

The observables of interest are two; the first one is the SINRD signature R_{SI} , defined as the ratio between the neutron count by the ^{238}U fission chamber (C_{FAST}) and the count differences in the Gd and Cd wrapped ^{239}Pu loaded fission chamber ($C_{Gd}-C_{Cd}$); the second one is the ratio between C_{FAST} and the count differences in the bare and the Cd wrapped a ^{235}U fission chambers (C_{TH}).

As explained in [14] the estimated uncertainty due to counting statistics for a measurement time of 1 h is strongly dependent on the burnup of the spent fuel assembly. This estimated uncertainty is lower than 5% for fuel with 3.5% initial enrichment, 5 years of cooling time, and burn-up larger than 15 GWd/t_{HM} [14]. The use of ^{239}Pu fission chambers enhances the sensitivity of the technique to ^{239}Pu content [11].

The two observables were estimated by means of Monte Carlo simulations with the code MCNPX 2.7.0 [15], the radionuclide composition of the fuel was taken from the spent fuel library developed at SCK•CEN [16,17,18]. Two observables, R_{SI} and C_{FAST}/C_{TH} , were determined for a total of 2940 cases of the spent fuel library [19].

The obtained results [7] for 17x17 PWR SFA indicated that SINRD can only be applied in dry conditions and that calibration curves can be determined to quantify ^{239}Pu provided that initial enrichment (IE) is known.

Figure 1 shows the R_{SI} observable as a function of ^{239}Pu content for different values of IE and BU. If the burnup (BU) is above 30 GWd/t_{HM} the data cluster around an almost straight line and there is a strong correlation between R_{SI} and ^{239}Pu content irrespective if the IE. For burnup below 30 GWd/t_{HM} the data exhibit a more scattered behaviour. This is due the fact that at lower BU values the presence of ^{235}U interferes due to the presence of a weak resonance at about the same energy as the one of ^{239}Pu .

The ratio C_{FAST}/C_{TH} , shown in Fig. 2, can be used to determine the IE if the BU is less than 30 GWd/t_{HM} and to account for the interference from ^{235}U . The ratio R_{SI} is almost independent from the cooling time (CT) up to CT of 300 years, while the ratio C_{FAST}/C_{TH} starts to decrease from not less than 10 years (Fig. 3).

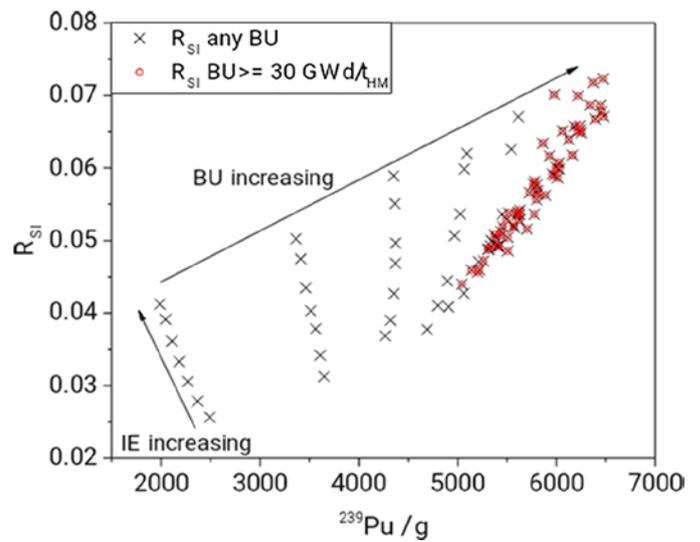


Figure 1: R_{SI} as function of ^{239}Pu amount for any BU values and for BU values of at least 30 GWd/t_{HM} . CT was 5 years and IE between 2% and 5%.

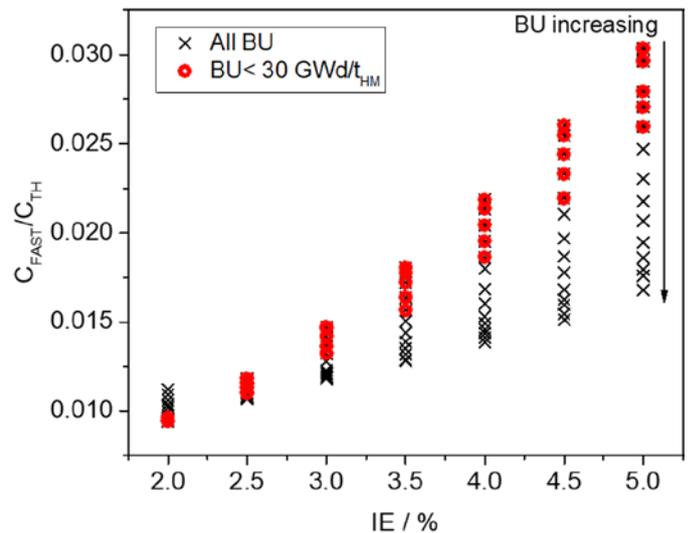


Figure 2: C_{FAST}/C_{TH} as a function of IE for any BU values and a CT of 5 years. The results for BU values of less than 30 GWd/t_{HM} are highlighted in a different colour.

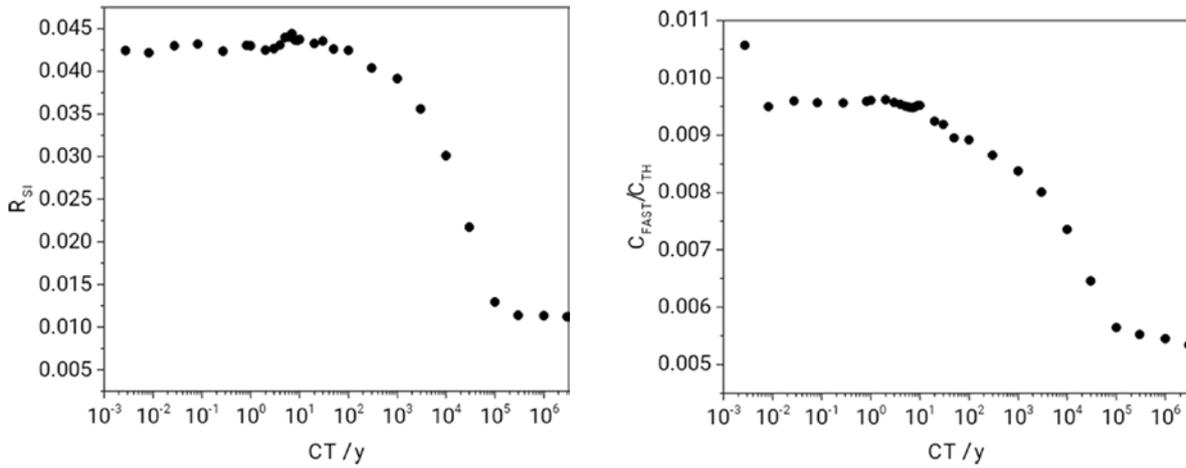


Figure 3: R_{Si} and C_{FAST}/C_{TH} as a function of CT. The data are for an IE of 2% and BU of 30 GWd/t_{HM}.

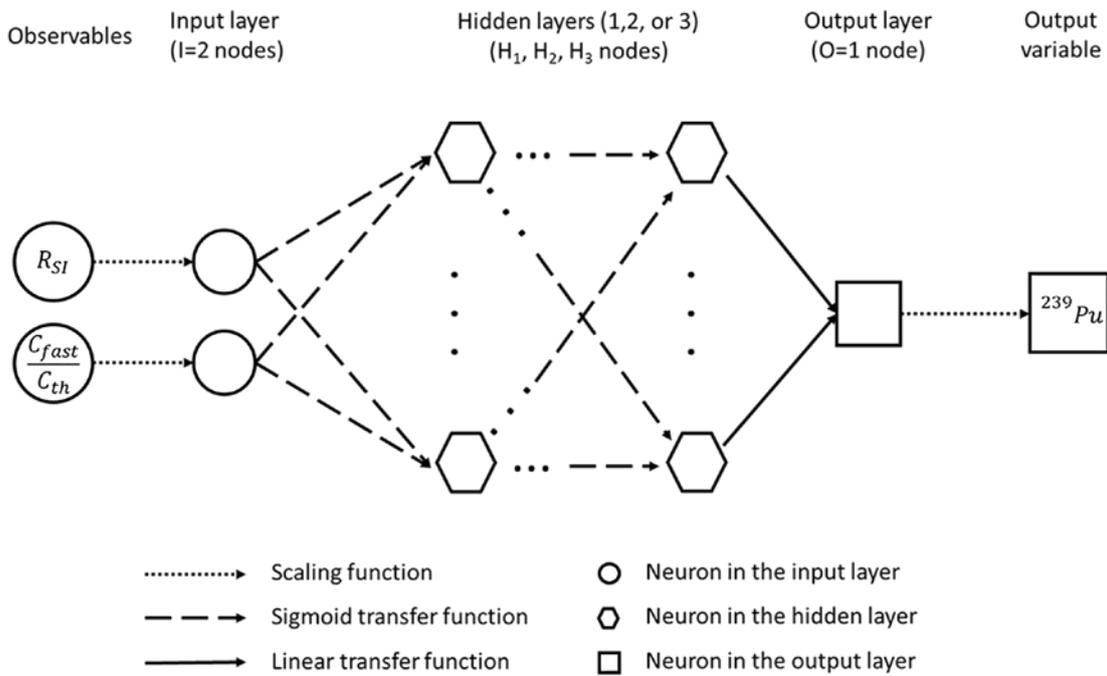


Figure 4: Architecture of the Neural network considered in this work.

3. Data analysis with artificial neural networks

3.1 Artificial Neural Networks

In addition to the above mentioned approach, we decided to investigate the use of artificial neural networks [12] as mean to determine the ²³⁹Pu quantity given the two observables R_{Si} and C_{FAST}/C_{TH} .

An ANN can be described as a network in which each node (or neuron) i processes the n input units it is connected to through a transfer (or activation) function f_i :

$$y_i = f_i \left(\sum_{j=1}^n (w_{ij} \cdot x_j - \theta_i) \right) \quad (1)$$

where y_i is the output of neuron i , x_j is the j -th input to node i , w_{ij} is the weight of the connection between input j and node i , and θ_i is the threshold (or bias) of the node. While each neuron i can have its own transfer function in our implementation [13, 20] the same transfer function was used for all the neurons in a given layer.

Neural networks have a multilayer architecture consisting of one layer for input neurons, one or more inner layers of neurons (also called hidden layers), and one layer for output neurons. The architecture chosen in this work is represented in Fig. 4 and consists up to three hidden layers. The number of nodes in the input layer is indicated with I , the number of nodes in the hidden layer number k is indicated with H_k , and the number of nodes in the output layer is indicated with O .

In this architecture, the input variables (Observables) are linearly scaled between -1 and +1 before being fed to the ANN. Sigmoid functions are used to connect the neurons in the input layer to the neurons in the first hidden layer, as well as to connect the neurons in the different hidden layers. Linear functions are used to connect the neurons in the last hidden layer to the neurons in the output layer. Finally the output from the neurons of the output layer is transformed into the output variable (^{239}Pu mass) with the inverse of the linear scaling function used for the considered observables.

The database of simulated observables and spent fuel characteristics is divided in three sets, corresponding to training, validation and testing. The training process of a neural network is an iterative process where the weights and biases of each neuron are adapted as result of the predictive error. Using an initial set of weights and biases, the training set is used by the training algorithm to calculate the predictive error and to adapt the weights and biases of the neurons. The predictive error is also computed for the validation set for each iteration during the training process. The predictive error for the training set and validation set normally decreases with the number of iterations but, as the network begins to overfit the dataset, the error in the validation set tends to increase. The weights and biases of the neurons in the trained network are those obtained for the minimum value of the predictive error in the validation set. The testing set is finally used to determine the accuracy of the trained ANN and to evaluate its capability to predict results from data it was not trained with. [21]

3.2 Data analysis of SINRD signatures

We considered only SINRD observables for cases with CT of 5 years, given the weak dependence on the CT up to a CT value of 10 years. The dataset is therefore reduced to 98 entries associated to 14 values of BU and 7 IE.

The size of the data set is relatively small when compared to the large data sets that are usually used when training an ANN; however, this is in line with the fact that, in view of a deployment of the ANN with experimental data, the data set would also not be large when considering realistic combinations of IE and BU values.

Starting from Eq. (1) we estimated the number of parameters of the ANN that need to be estimated during the training procedure as a function of the network structure. The results are shown in Table 1 and indicate the number N of parameters as a function of the number of nodes I in the input layer, the number of nodes H_k in the hidden layer number k , and the number of nodes O in the output layer. Different configurations with k between one and three were considered with a minimum level of complexity, in view of the limited availability of training data.

In this work we use an architecture where the two observables enter the input layer, and the ^{239}Pu amount is the only quantity in the output layer. Therefore for our case $I=2$, representing the two SINRD observables, and $O=1$, representing the ^{239}Pu amount.

In practice, the data processing with ANN consists in identifying a ANN configuration that describes the dependence of ^{239}Pu varies in the space of the variable $C_{\text{FAST}}/C_{\text{TH}}$ and R_{SI} .

I	H ₁	H ₂	H ₃	O	N
2	3			1	13
2	2	2		1	15
2	2	2	2	1	21

Table 1: Number n of the ANN parameters to be determined as a function of the network architecture.

For a given ANN configuration, there is a level of arbitrariness when choosing the size of training, testing and validation. In our case, given the limited size of the whole database and the limited possibilities to carry out actual measurement we decided to limit the size of the training and validation set to 10 entries each; the rest of the database was used for testing.

Since the number N of the network parameters should not exceed the size of the training and validation data set we opted for a network with two hidden layers with two nodes each. The choice of two layers stems from previous experience where we learned that better performance can be achieved when the number of layers is increased [13, 20].

The analysis of the data was carried with a tool [20] developed in MATLAB [22]; the tool allows to carry out the analysis through a graphical user interface (GUI); through this GUI the user can import data from a text file, filter the data based on certain criteria, define the network architecture and several optimization options such as internal processing functions, performance function and training function [20]. The results and network configuration can be exported. The tool allows training with random or fixed initial values for weights and offset. Also the entries of the database to be used for training, validation can be chosen randomly by the programme or by the user via flags associated to entries in the database.

In our analysis, the optimization of the ANN through the quantity mean square error (mse)

$$mse = \frac{1}{N} \sum_{k=1}^N (A_{k,calc} - A_k)^2 \quad (2)$$

Where $A_{k,calc}$ is the value of the parameter as determined by the ANN in the output layer, A_k is the nominal value of

the parameter. After each iteration (also called epoch) the weights and biases of the nodes were updated according to the Levenberg-Marquardt [23,24,25] or Bayesian regularization [26,27] algorithms as implemented in Matlab with the *trainlm* and *trainbr* functions, respectively. The value of the parameters used in the both algorithms are included in Table 2.

Maximum number of epochs	10 ⁵
Maximum time to train (in seconds)	1000
Network performance goal	0
Minimum performance gradient	10 ⁻⁷
Maximum value for the Marquardt adjustment parameter (mu)	10 ¹⁰

Table 2: Parameters used for the training algorithms.

In our analysis we tested the option not to include the scaling function described in Fig. 4 between observables and input layer and between output layer and output variables. This test resulted in worse values of the *mse* when compared with an analysis that include this scaling step.

3.3 Results

Initially we kept random initial values for weights and offset as well as the entries of the database to be used for training, testing and validation. We observed a large variation in the number of epochs before the training timing ended as well as a variation of several orders of magnitude in terms of *mse*.

We realized that it is impossible to identify the configuration with the minimum *mse* based on a random or sequential selection of the database entries. Assuming that the assignment of an entry to the validation or training data set does not matter, with 20 entries in 98 this would correspond to

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{98!}{20!(78)!} \sim 3 \times 10^{20}$$

combinations.

With the used data analysis tool, both the initial values for weights and offset as well as the entries for training and testing can be defined by the user or chosen randomly.

Since weights and offset define the ANN to be trained, the choice of the initial values of weights and offset was kept random.

We then tried to identify, by expert judgement and hence natural intelligence, the entries of the database. The choice of the entries was carried out by studying how the ²³⁹Pu varies in the space of the variable C_{FAST}/C_{TH} and R_{SI} . This is shown in Fig. 5.

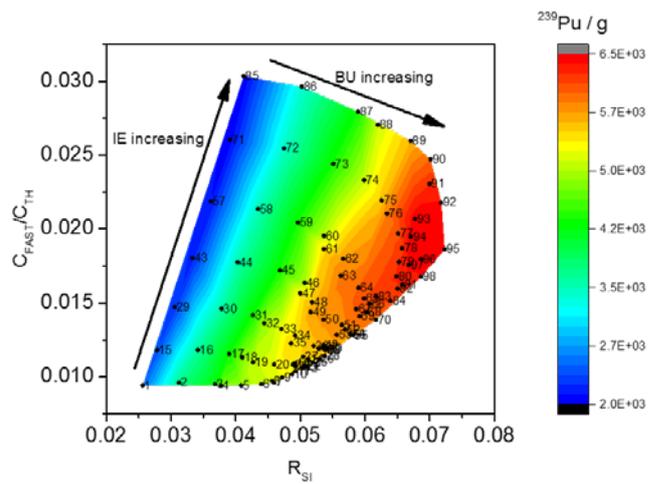


Figure 5: ²³⁹Pu amount as a function of R_{SI} and C_{FAST}/C_{TH} . The number indicate the corresponding entry in the considered data set.

The data in Fig. 5 reveal that C_{FAST}/C_{TH} and R_{SI} have a limited range of variation and lie in specific domain. As explained before, we are looking for a ANN configuration that describes the dependence of ²³⁹Pu in the space of the variables C_{FAST}/C_{TH} and R_{SI} . Given the results in Fig. 5, it was logical to assume that entries to use in the training and validation data set should lie at the boundary of the domain of C_{FAST}/C_{TH} and R_{SI} . In addition, it seemed logical that a sufficient number of them should lie inside the area in order to allow to describe the shape over the domain of C_{FAST}/C_{TH} and R_{SI} .

Based on these criteria, we defined the entries for the training validation data set as indicated in the Table 3 and Fig. 6. It is worth to comment that the entries indicated in italic, although present in the spent fuel library, are not realistic since the BU is too high for the chosen IE. These entries were nevertheless considered to assess the performance of the method in a configuration where all the entries of the database of spent fuel library observables can be used.

The training was then repeated one hundred times to try different initial values for weights and offset.

It was found initial values for weights and offset can affect the *mse* up to two order of magnitude; this corresponds to about 1 order of magnitude change in the resulting ²³⁹Pu mass standard deviation.

The best obtained value of *mse* was 2.3×10^4 . The square root of *mse* corresponds to a standard deviation between declared and predicted of ²³⁹Pu about 150 g. Changing the training function between *trainlm* and *trainbr* did not seem to affect the results.

The deviation between the predicted ²³⁹Pu mass and the value in the data base is shown in Fig. 7, where the entries used for training, validation and testing are shown with different colours.

ID	Flag	BU	IE	CT	ID	Flag	BU	IE	CT
1	T	5	2	5	5	V	25	2	5
28	T	70	2.5	5	29	V	5	3	5
31	T	15	3	5	35	V	35	3	5
59	T	15	4	5	45	V	15	3.5	5
63	T	35	4	5	57	V	5	4	5
75	T	25	4.5	5	64	V	40	4	5
78	T	40	4.5	5	70	V	70	4	5
85	T	5	5	5	87	V	15	5	5
89	T	25	5	5	91	V	35	5	5
95	T	55	5	5	93	V	45	5	5

Table 3: Selected entries for training (T) and validation (V) data sets.

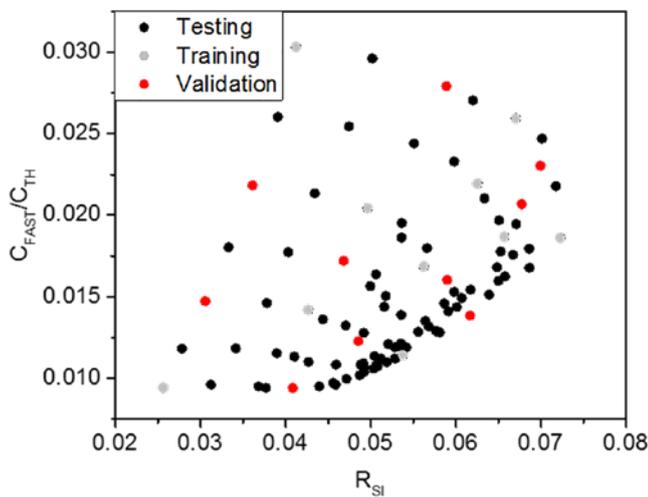


Figure 6: Training, validation and testing data sets.

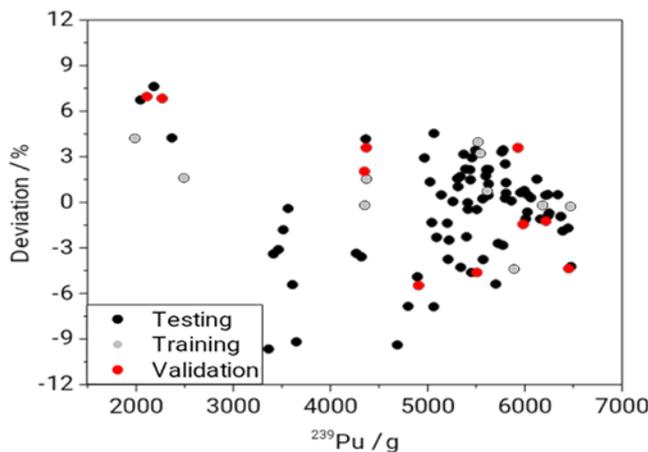


Figure 7: Percentage deviation between predicted and nominal ^{239}Pu amount for training, validation and testing data sets.

In the testing data set, the average deviation between the nominal ^{239}Pu content and the calculated one was 0.2% with a standard deviation of 3.5% and a maximum deviation of 10%.

4. Conclusions

We described a data analysis approach based on artificial neural networks (ANN) to process the observables associated to the SINRD technique. The SINRD observables were obtained with Monte Carlo based simulations using fuel composition from a spent fuel library and represent a data set of nearly 3,000 entries.

Given the fact that the SINRD observables are not depending on cooling time up to a cooling time of 10 years, we restricted the analysis to the 98 entries with a cooling time of 5 years.

While the choice of the initial values of weights and offset was kept random, we identified by expert judgement and hence natural intelligence, the 20 entries of the database to be used for training and validation. The obtained results reveal that, the average deviation between the nominal ^{239}Pu content and the calculated one was 0.2% with a standard deviation of 3.5% and a maximum deviation of 10%.

The selection of spent fuel assemblies based on expert judgement is not necessarily the best, in terms of ANN precision, but allowed to resolve quickly a problem that would have not been possible to solve by selecting on a randomly the database entries for training and validation.

Future work will focus on possibly reducing even further the size of the data base for training and validation and limiting only on realistic cases of the library of observables. In addition, we would like to study a different form of the performance function for example accounting for the relative deviations rather than absolute deviations. We would like also to analyse data with different cooling times by including an additional observables from the observable data-base in the data analysis.

5. References

- [1] D. Reilly, N. Ensslin, H. Smith Jr. and S. Kreiner, *Passive Nondestructive Assay of Nuclear Materials*, NUREG/CR-5550, LA-UR-90-732, 1991
- [2] H. Würz, *A Simple Non-Destructive Measurement System For Spent Fuel Management*, Nuclear Technology 95 (1991) pp 193-206
- [3] Park W. S., et al., 2014. "Safeguards by design at the encapsulation plant in Finland". Proceedings of the 2014 IAEA safeguards symposium.
- [4] European Council Decision 2006/976/Euratom of 19 December 2006
- [5] *Coordinated Technical Research Meeting on Spent Fuel Verification Methods*, IAEA March 3-6 2003
- [6] A. Borella, et al., *Advances in the Development of a Spent Fuel Measurement Device in Belgian Nuclear Power Plants.*- In: Symposium on International Safeguards, Linking Strategy, Implementation and People, Book of Abstracts, Presentations and Papers, 2014, IAEA, Vienna, Austria, p. 272
- [7] R. Rossa, *Advanced non-destructive methods for criticality safety and safeguards of used nuclear fuel*, PhD thesis at Université libre de Bruxelles, September 2016.
- [8] S. Tobin, H. Menlove, M. Swinhoe, M. Schear, *Next Generation Safeguards Initiative research to determine the Pu mass in spent fuel assemblies: Purpose, approach, constraints, implementation, and calibration*, Nuclear Instruments & Methods In Physics Research Section A, 2011, doi 10.1016/j.nima.2010.09.064
- [9] T. White, et al., 2018. "Application of Passive Gamma Emission Tomography (PGET) to the inspection of spent nuclear fuel". Proceedings of the 2018 INMM annual meeting.
- [10] H. Menlove, C. Tesche, M. Thorpe and B. Walton, *A Resonance Self-Indication Technique for Isotopic Assay of Fissile Materials*, Nuclear application Vol. 6 April 1969 pp 401-408
- [11] Rossa, R., Borella, A. & van der Meer, K. *Investigation of the Self-Interrogation Neutron Resonance Densitometry applied to spent fuel using Monte Carlo simulations* Jan 2015 In : Annals of Nuclear Energy. 75, p. 176-183
- [12] Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S., *Multilayer feedforward networks with a nonpolynomial activation function can approximate any function*, Neural networks, 6(6):861-867, 1993.
- [13] Borella, A., Rossa, R. & Turcanu, C, *Signatures from the spent fuel: simulations and interpretation of the data with neural network analysis*, 1 Dec 2017 In : ESARDA Bulletin. p. 29-38 10 p.
- [14] Rossa R., Borella A., Labeau P.E., Pauly N., van der Meer K., Neutron absorbers and detector types for spent fuel verification using the self-interrogation neutron resonance densitometry, Nuclear Instruments and Methods in Physics Research A 791 (2015), 93-100.
- [15] D.B. Pelowitz, Ed., "MCNPX Users Manual Version 2.7.0" LA-CP-11-00438 (2011).
- [16] R. Rossa., et al., *Development of a reference spent fuel library of 17x17 PWR fuel assemblies*, ESARDA BULLETIN, No. 50, December 2013
- [17] A. Borella, et al., *Extension of the SCK•CEN spent fuel inventory library*, 37th ESARDA Annual Meeting, 2015
- [18] A. Borella, M. Gad, R. Rossa, K. van der Meer, *Sensitivity Studies on the Neutron Emission of Spent Nuclear Fuel by Means of the Origen-ARP Code*, In: Proceeding of the 55th INMM Annual Meeting, Atlanta, Georgia, United States, July 2014.
- [19] Borella, A., Rossa, R. & van der Meer, K. *Simulated observables for spent fuel non-destructive assay* 22 Jul 2018 Proceedings of the 2018 INMM annual meeting
- [20] Zaïoun, H. *Analysis and interpretation of spent fuel signatures with an artificial neural network: Improvement of the network, based on the previous work, and development of a graphical user interface* 30 Mar 2018 Studiecentrum voor Kernenergie. 25 p. (SCK•CEN Reports; no. I-0687)
- [21] <https://nl.mathworks.com/help/deeplearning/ug/divide-data-for-optimal-neural-network-training.html>
- [22] <https://www.mathworks.com/help/matlab/>
- [23] <https://nl.mathworks.com/help/deeplearning/ref/trainlm.html>
- [24] Marquardt, D., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM Journal on Applied Mathematics, Vol. 11, No. 2, June 1963, pp. 431-441
- [25] Hagan, M.T., and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, Vol. 5, No. 6, 1999, pp. 989-993, 1994.
- [26] MacKay, David J. C. "Bayesian interpolation." Neural computation. Vol. 4, No. 3, 1992, pp. 415-447.
- [27] Foresee, F. Dan, and Martin T. Hagan. "Gauss-Newton approximation to Bayesian learning." Proceedings of the International Joint Conference on Neural Networks, June, 1997.