Inferring initial enrichment, burnup, and cooling time of spent fuel assemblies using artificial neural networks

Riccardo Rossa, Alessandro Borella

SCK CEN Belgian Nuclear Research Centre
Boeretang 200, Mol 2400 Belgium
E-mail: rrossa@sckcen.be

Abstract

The verification of spent nuclear fuel is a major task during a safeguards inspection and inspectors have to verify both the correctness and completeness of the operator declaration. The traditional way to verify spent fuel is with non-destructive assays (NDA) relying on the radiation emission from the fuel. The NDA measurement results are then compared with estimates based on operator declaration of initial enrichment, burnup, and cooling time.

However, the radiation emission from spent fuel is affected by the fuel parameters and irradiation history, and research is ongoing to improve the data analysis of NDA measurement results. In this work artificial neural networks were developed to infer the initial enrichment, burnup, and cooling time of spent fuel assemblies from simulated NDA measurements with the Forkball instrument.

Several neural networks architectures and detector responses were compared to find the optimal network configuration to infer the spent fuel parameters. Results show that the cooling time is the most challenging parameter to estimate and the associated data processing step plays a crucial role in its reliable estimate. The combination of multiple detector responses also leads to a significant improvement in the determination of the initial enrichment, burnup, and cooling time. The optimal neural networks in this study are able to determine the initial enrichment and burnup within 12%, and the cooling time, using the data processing step, within 4%.

Keywords: neural networks; machine learning; spent fuel; NDA; initial enrichment; burnup; cooling time

1. Introduction

The International Atomic Energy Agency (IAEA) has the legal mandate under the Non-Proliferation Treaty (NPT) [1] to verify the nuclear material inventory in the States party to the treaty. In 2020 safeguards were applied in 183 countries and in more than 700 facilities [2]. Nuclear power reactors represent a significant share of the facilities under safeguards, and most of the fissile material inventory is in the form of irradiated or spent fuel.

The goal of a safeguards inspection is to verify both the correctness and completeness of the declaration given by the operator to the IAEA. In the case of spent fuel verification, non-destructive assay (NDA) is generally used to measure the radiation emitted by spent fuel. The Digital Cherenkov Viewing Device (DCVD) and the Fork detector are among the most used NDA for spent fuel verification. However, these NDA instruments measure mostly emissions due to fission products (e.g. ^{137}Cs) and minor actinides (e.g. ^{244}Cm) since they are the main gamma-ray and neutron emitters, respectively [3].

Therefore, to verify the inventory of nuclear material (i.e. ^{235}U and Pu), the NDA measurement results are compared with calculations that rely on operator declarations of fuel irradiation history (e.g. initial enrichment, burnup, and cooling time). Recent studies [4] showed that the fuel irradiation history significantly impacts the radiation emission and research is ongoing to improve the current data analysis approach.

Due to the complexity of the fuel composition and multivariate nature of the NDA measurement results, machine learning is increasingly applied in the field of spent fuel verification [5],[6],[7],[8]. Artificial Neural Networks (ANNs) are being increasingly chosen for the data analysis of safeguards tasks, such as the image analysis and surveillance review [9],[10],[11],[12],[13],[14]. In the field of spent fuel verification, ANNs still have a rather limited application for the estimation of the spent fuel parameters [15],[16]. The ANNs developed for the different safeguards tasks vary greatly in terms of network size and network configuration.

In this work, ANNs were developed using as inputs the detector responses from the Forkball detector [17] with the aim of inferring the initial enrichment (IE), burnup (BU), and

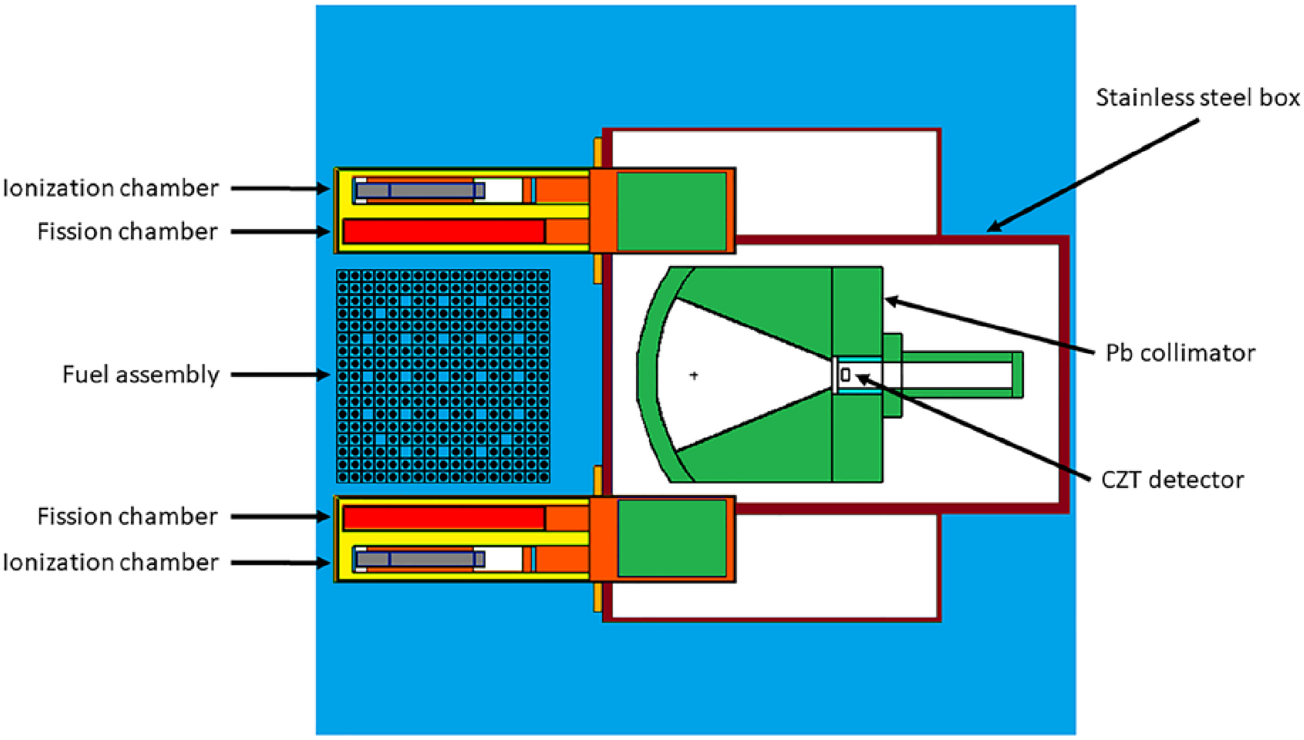

Figure 1: 2-D view of the Monte Carlo model of the Forkball detector measuring a spent fuel assembly. Figure taken from [6].

cooling time (CT) of the spent fuel assembly. A separate ANN was developed for each output parameter and different ANN architectures were compared in terms of mean absolute percentage error.

Previous work [15] published using ANNs for the estimation of spent fuel parameters focused mostly on the training accuracy and considered small ANNs architectures with maximum of 20 neurons per layer. The ANNs presented in this article investigate the effect of data processing of the output features and include a detailed discussion on the ANNs performance.

The dataset used for the study is introduced in Section 2, whereas the ANNs basic principles and architectures are described in Section 3. The results from the developed ANNs are presented in Section 4 and are followed by discussion in Section 5. The conclusions from the study are summarized in Section 6.

2. Dataset

A dataset containing the detector responses from the Forkball detector was used for the development of the ANNs in this study. The Forkball detector is an NDA instrument being conceived for underwater measurement of spent fuel and combines the detector responses of the Fork detector (e.g. total neutron count from the fission chambers, current from the ionization chambers) with the gamma-ray spectroscopic capabilities of a Cadmium Zinc Telluride (CZT) detector [17]. The Forkball detector is made up of two polyethylene arms each containing one fission chamber and one ionization chamber, connected by a large Pb shielding and collimator that hosts the CZT detector. During the measurement, the spent fuel assembly is placed between the two polyethylene arms as shown in Figure 1.

A total of 1960 Monte Carlo simulations were carried out with the MCNPX code [18] to compute the Forkball detector responses for fuel assemblies with a wide range of initial enrichment, burnup, and cooling time. The approach for the calculation of the detector responses is described in [15], and an extract of the dataset is shown in Table 1. The detector responses were taken as input features of the ANNs whereas either the initial enrichment, burnup, or cooling time was taken as output feature of the ANNs. The initial enrichment ranged from 2.0% to 5.0% in steps of 0.5%, the burnup ranged from 5 GWd/t_{HM} to 70 GWd/t_{HM} in steps of 5 GWd/t_{HM}, and the cooling time ranged from 1 day to 100 years with 18 intermediate values. Since the cooling time values were logarithmically separated, the variable CT' was also considered as output feature

$$CT' = \ln(cooling\ time) + 10 \tag{1}$$

Input features							Output features			
Fission chamber		Ionization chamber	Cadmium Zinc Telluride							
Total neutrons (cps)	Fast neutrons (cps)	Current (nA)	^{134}Cs, 605 keV (cps)	137Cs, 662 keV (cps)	^{134}Cs, 796 keV (cps)	^{154}Eu, 1274 keV (cps)	IE (%)	BU (GWd/t)	CT (y)	CT' (ln(y))
1.4	0.6	154.8	85.0	466.4	147.4	4.4	2.0	5	1	10.00
1.2	0.5	12.7	11.3	406.0	19.6	2.7	2.0	5	7	11.95
1.0	0.4	6.5	<0.1	238.8	<0.1	0.3	3.5	5	30	13.40
10.2	4.7	42.1	73.4	1207.4	127.3	17.8	4.0	15	7	11.95
9.9	4.5	5.1	<0.1	186.8	<0.1	<0.1	2.5	20	100	14.61
108.4	49.8	71.9	146.2	1950.4	253.5	49.4	3.5	25	8	12.08
1142.6	524.7	634.4	3060.9	3174.5	5308.4	172.9	2.5	35	1	10.00
110.0	51.5	32.1	<0.1	1166.3	<0.1	3.6	4.5	40	50	13.91
108.4	49.8	71.9	146.2	1950.4	253.5	49.4	3.5	25	8	12.08
1142.6	524.7	634.4	3060.9	3174.5	5308.4	172.9	2.5	35	1	10.00
110	51.5	32.1	0	1166.3	0	3.6	4.5	40	50	13.91

Table 1: Extract of the dataset containing the simulated detector responses of the Forkball (input features) and the corresponding initial enrichment (IE), burnup (BU), cooling time (CT), and CT' (output features). The detector responses included in the table are rounded to the first decimal digit.

3. Artificial neural networks

3.1 Basic principles

ANNs are a subset of machine learning models that aim to replicate with mathematical functions the neurons in a biological brain. ANNs can be used as universal function approximators [19] and are being developed for a wide range of applications such as pattern recognition [20], data mining [21], and cyber security [22]. In the nuclear field ANNs have been used recently for example in gamma-ray spectroscopy [23], severe accident analysis [24], and nuclear medicine [25].

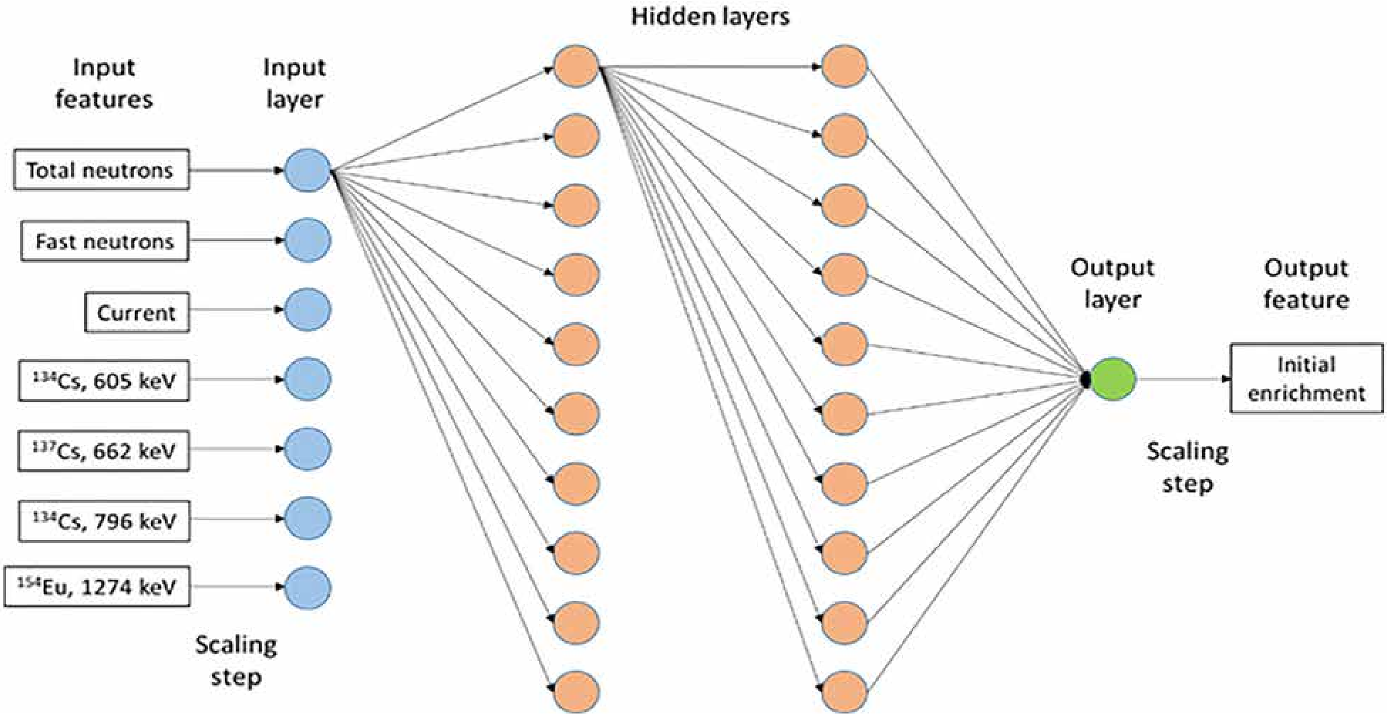

Figure 2: example of ANN architecture used for this study. The ANN includes all input features, two hidden layers each with 10 neurons, and initial enrichment as output variable. The ANN is fully connected but for graphical reasons not all connections among neurons are included in the figure.

The building unit of the ANN is the artificial neuron which, as the biological neuron, receives input signals and through an activation function and bias produces an output signal [19].

The general structure of an ANN (also called network architecture) is shown in Figure 2. The dataset observables are first connected to the so-called input layer via a scaling function that is usually recommended to improve the model accuracy and speed of convergence [26]. The input layer is followed by one or more hidden layers and finally by an output layer. The output of the output layer may be again re-scaled to obtain the desired output variables. Neurons in one layer can be either connected to all neurons in the following layer (so-called fully connected networks), or a group of neurons can be connected only to one neuron in the following layer (so-called pooling networks) [27].

Fully connected ANNs are used for a broad range of applications, whereas pooling networks are generally used for image analysis [28]. Fully connected ANNs were developed in this work since they do not require any assumption to be made on the input features.

The general equation for an artificial neuron is:

$$y_i = f_i\left(\sum_{j=1}^{N}\left(w_{i,j} \cdot x_i + b_i\right)\right) \tag{2}$$

Where y_i is the neuron output, f_i is the activation function, N is the number of input neurons to neuron i, $w_{i,j}$ is the weight of the connection between input neuron j and neuron i, x_i is the neuron input, and b_i is the bias for neuron i. In case of fully connected ANNs N is the same for each neuron in one layer.

The development of an ANN model can be divided into a training phase and a prediction phase. The observations in the dataset are randomly divided thus into a training dataset and testing dataset. The weights and biases of each neuron are initialized with random values at the start of the training phase, and then are optimized according to a loss function and optimization function defined by the user. The activation function for each layer and the number of iterations (also called epochs) performed during the training phase are also specified by the user. The weights and biases optimized during the training phase are finally used in the prediction phase on the observations in the testing dataset

3.2 Network architecture

Several ANN architectures were developed and compared in this study. During the development of the ANNs the observations in the full dataset have been randomly divided into training dataset (70% of observations) and testing dataset (30% of observations). The training dataset was further split into 5 folds to perform a k-fold cross-validation analysis [29].

ANNs were developed considering one input feature (e.g. total neutron count) only or combining all available input features from the Forkball. Before entering the ANN the input features were scaled to a distribution centred around 0 and with standard deviation of 1. The scaling factors are determined using the training dataset and the scaling is then applied to both training and testing datasets.

Both one and two hidden layers were considered, with the number of neurons ranging from 10 to 500 and Relu [30] activation functions for the hidden layers. The network optimization was carried out using the mean absolute percentage error ($mape$) as loss function and the ADAM [31] optimizer with 10^{-3} learning rate. The $mape$ loss function was calculated according to the formula [32]:

$$mape(y_{true}, y_{pred}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \frac{\left|y_{true,i} - y_{pred,i}\right|}{max\left(\epsilon, \left|y_{true,i}\right|\right)} \tag{3}$$

Where y_{true} is the true value of the $i\text{-}th$ sample, y_{pred} is the corresponding predicted value, $n_{samples}$ is the number of samples, and ϵ is an arbitrary small yet strictly positive number to avoid undefined results when y_{true} is zero. The $mape$ loss function was chosen as metric for the ANNs performance because it is sensitive to the relative errors rather than the global scaling of the output features.

The optimization phase was carried out for 100 epochs and the algorithm convergence was verified by plotting the loss function as a function of the number of epochs.

For each ANN architecture, the split of the dataset into training and testing datasets was repeated 10 times and each time the $mape$ was recorded. The ANN performance was finally calculated as the average and standard deviation of the $mape$ for the training, validation, and testing datasets over the 10 repetitions. Similar results were obtained for the calculated $mape$, therefore only the values related to the testing datasets are reported in the paper.

4. Results

4.1 One hidden layer ANN

ANNs with one hidden layer were developed and the number of neurons in the hidden layer was taken as the only hyper-parameter for optimization of the network architecture. The comparison between the output parameter of the ANN and the declared value is shown in Figure 3 for initial enrichment and burnup. The same comparison is shown in Figure 4 for cooling time and CT'. The $mape$ of

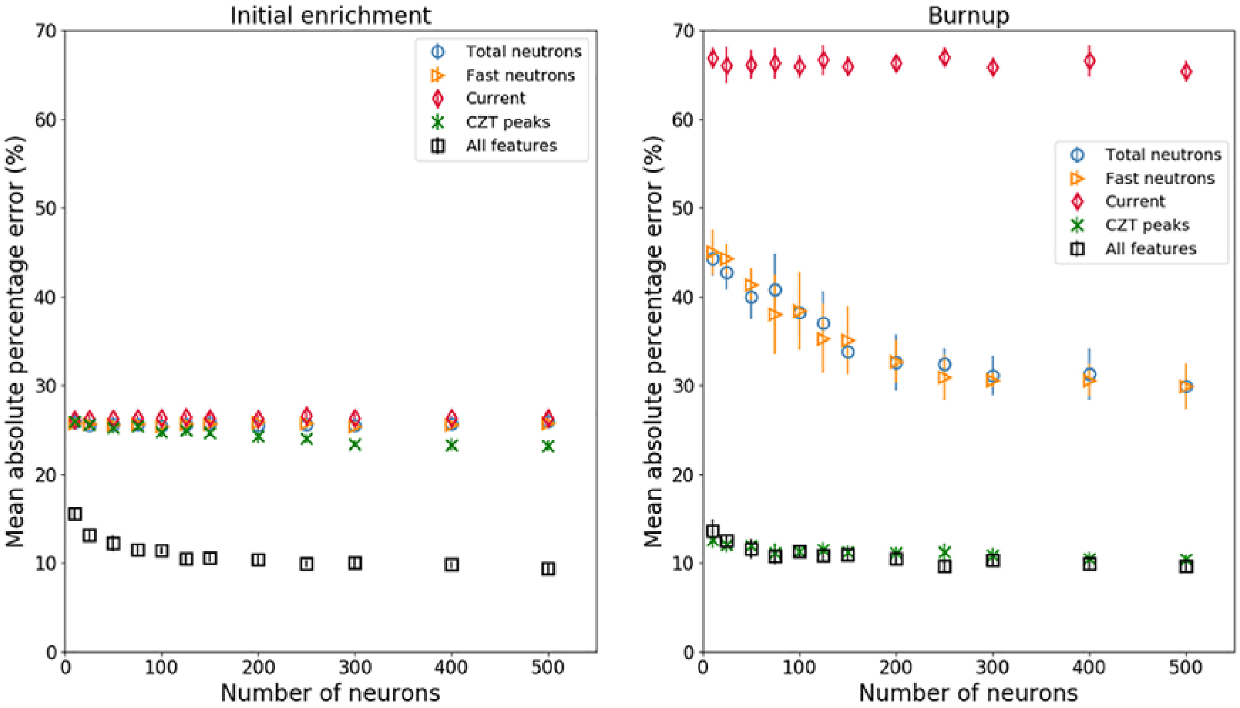

Figure 3: Mean absolute percentage error for the determination of the initial enrichment (left) and burnup (right). The values refer to ANNs with one hidden layer and the mape is shown as a function of the number of neurons in the ANN.

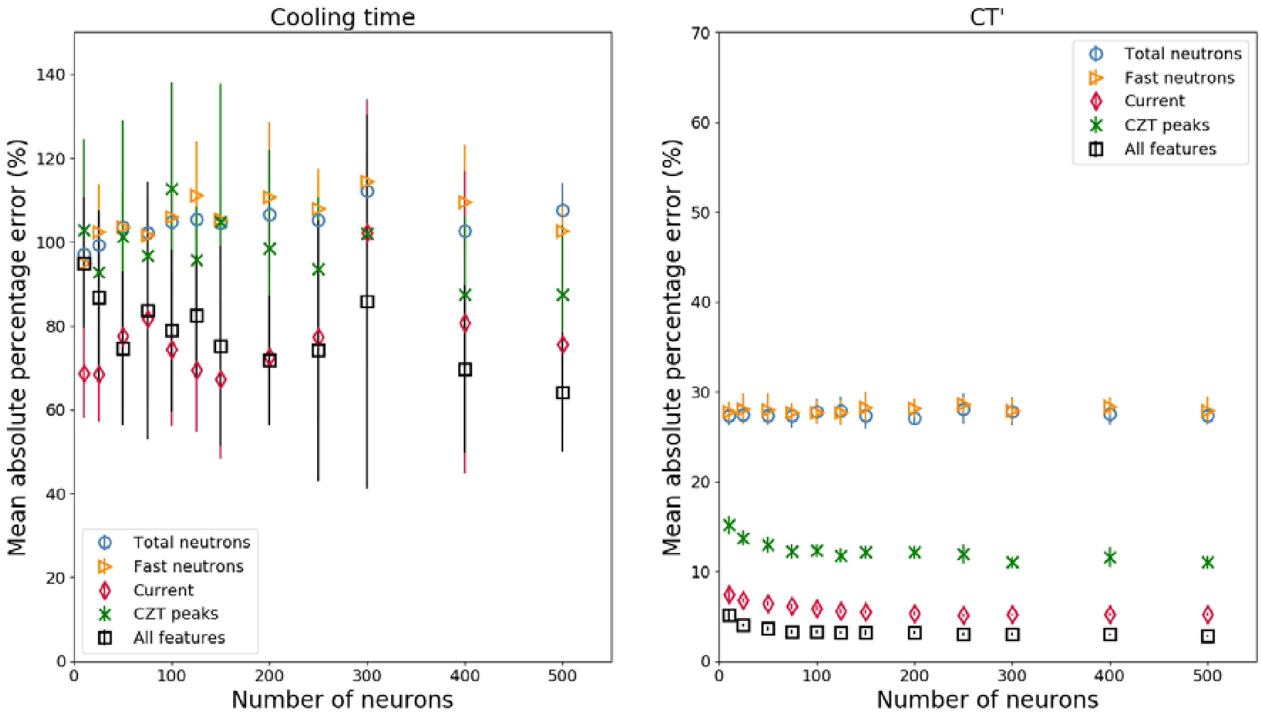

Figure 4: Mean absolute percentage error for the determination of the cooling time (left) and CT' (right). The values refer to ANNs with one hidden layer and the mape is shown as a function of the number of neurons in the ANN.

the ANNs is shown as a function of the number of neurons in the hidden layer.

The results for the determination of the initial enrichment show that by using one feature only as input to the ANN, the error on the estimated initial enrichment is about 25% with almost no appreciable dependence on the number of neurons in the hidden layer. However, combining all features, the $mape$ of the ANN estimate decreases with the number of neurons from 15.5% with 10 neurons down to

9.4% with 500 neurons. It is observed that the reduction in $mape$ is larger up to 125 neurons in the ANN ($mape$ of 10.5%) and remains rather stable by further increasing the number of neurons. The uncertainty associated to the $mape$ due to the random selection of observations in the training dataset is for all cases within 1%. The uncertainty mentioned here and in the rest of the paper is resulting from the 10 iterations of the random splitting of observations in the training and testing datasets.

In case of burnup determination, by using the fission chamber features individually (i.e. Total neutrons, Fast neutrons) the $mape$ decreases with the number of neurons from 45% with 10 neurons down to 30% with 500 neurons. The $mape$ using the ionization chamber feature (Current) remains around 66% independently from the number of neurons, whereas using the gamma-ray spectroscopy features the $mape$ slightly decreases with the number of neurons from 12.6% with 10 neurons down to 10.4% with 500 neurons. By combining all features, the $mape$ ranges from 13.6% with 10 neurons to 9.6% with 500 neurons. As in the case of the initial enrichment determination, the largest decrease in $mape$ was observed increasing the ANN size up to 125 neurons. The uncertainty associated to the $mape$ due to the selection of observations in the training dataset is within 4% when using fission chamber features and within 2% using the ionization chamber feature. The uncertainty decreases to within 1% in almost all cases by using the gamma-ray spectroscopy features or combining all features in the ANN.

The results for the determination of the cooling time show that the $mape$ is for many cases above 100% and significantly larger compared to the determination of the initial enrichment and burnup. Therefore, in the rest of the paper, the cooling time variable was not considered anymore in the analysis. Compared to the cooling time variable, the $mape$ values for CT' significantly improve and are between 27-28% using the fission chamber features, between 7.4% and 5.1% using the ionization chamber feature, between 15.2% and 11% using the gamma-ray spectroscopy features, and between 5.1% and 2.8% combining all features. The $mape$ for the CT' feature is expressed in years once the feature has been transformed using the inverse function of Formula (1). Apart from the case of using only fission chamber features, the $mape$ decreases by increasing the size of the ANN, with the most significant decrease with ANN of up to 125 neurons. The uncertainty associated to the $mape$ due to the selection of observations in the training dataset is within 2% when using only fission chamber features, and within 1% in all other cases.

The cooling time values available in the dataset are logarithmically spaced and the transformation of Formula (1) allows to define the CT' variable with linearly spaced values. This feature transformation leads to a strong reduction of the $mape$ for the ANNs using the CT' feature and is another evidence of the importance of data processing in case of variables that have a large range of values.

The other output features in the dataset are not transformed using equivalents of Formula (1) since they are already linearly spaced in the dataset.

4.2 Two hidden layers ANN

ANNs with two hidden layers were developed by using the features of the Forkball instrument and the number of neurons in both hidden layers as hyper-parameters. The $mape$ for the determination of the spent fuel parameters is shown in Table 2 for ANN models using all features available for the Forkball instrument. Results for models using only one feature are not included since they obtained significantly larger $mape$. The $mape$ for the determination of the CT' feature is expressed in years once the feature has been transformed using the inverse function of Formula (1). Results using the cooling time as output feature for the ANNs are not included since the estimates showed unreliable results as in the previous section.

The $mape$ for the initial enrichment estimate shows a limited decrease from 13.2% in the case of ANN with 10 neurons in each hidden layer to 8.9% for ANN with 75 neurons in each hidden layer. However, the $mape$ does not decrease further by increasing the number of neurons in any of the hidden layers and reaches 8.7% in the case of ANN with 500 neurons in each hidden layer. The results suggest that the choice of the hidden layer to be increased in size is not crucial, but slightly smaller $mape$ were obtained with ANNs with equal number of neurons in each hidden layer. The uncertainty associated to the $mape$ due to the selection of observations in the training dataset is within 1% for almost all ANN models.

The $mape$ for the burnup estimate also shows a limited decrease from 11.4% in the case of ANN with 10 neurons in each hidden layer to 8.4% for ANN with 75 neurons in each hidden layer. As for the initial enrichment estimation, the $mape$ does not decrease further by increasing the number of neurons in any of the hidden layers and reaches 7.7% in the case of ANN with 500 neurons in each hidden layer. The results suggest that the choice of the hidden layer to be increased in size is not crucial, but slightly smaller $mape$ were obtained with ANNs with equal number of neurons in each hidden layer. The uncertainty associated to the $mape$ due to the selection of observations in the training dataset is within 1% for almost all ANN models.

The $mape$ for the CT' estimate shows a limited decrease from 3.9% in the case of ANN with 10 neurons in each hidden layer to 2.4% for ANN with 500 neurons in each hidden layer. The uncertainty associated to the $mape$ due to the selection of observations in the training dataset is within 0.5% for almost all ANN models.

A		Initial enrichment - Number of neurons in the second hidden layer											
		10	25	50	75	100	125	150	200	250	300	400	500
Number of neuron in the first hidden layer	10	13.2 ± 0.7	12.4 ± 1.0	11.7 ± 0.5	11.3 ± 0.8	11.0 ± 1.0	10.3 ± 0.6	10.5 ± 0.7	10.3 ± 0.5	10.7 ± 0.7	10.5 ± 0.6	9.8 ± 0.4	9.9 ± 0.6
	25	11.4 ± 0.9	10.4 ± 0.5	10.4 ± 0.5	10.3 ± 0.6	9.7 ± 0.9	9.3 ± 0.4	9.5 ± 0.6	9.5 ± 0.4	9.2 ± 0.4	9.0 ± 0.7	9.3 ± 0.5	8.9 ± 0.6
	50	10.1 ± 0.6	9.8 ± 0.6	9.4 ± 0.5	9.1 ± 0.4	9.3 ± 0.5	9.3 ± 0.8	9.4 ± 0.8	9.0 ± 0.7	8.7 ± 0.8	9.0 ± 0.6	8.9 ± 0.7	8.3 ± 0.5
	75	9.6 ± 0.6	9.2 ± 0.7	9.1 ± 0.4	8.9 ± 0.6	8.8 ± 0.5	8.5 ± 0.7	8.6 ± 0.8	8.4 ± 0.5	8.6 ± 0.6	8.5 ± 0.7	8.5 ± 0.9	8.5 ± 0.7
	100	9.5 ± 0.8	9.1 ± 0.5	9.1 ± 0.5	9.1 ± 0.6	8.7 ± 0.6	8.9 ± 0.7	8.6 ± 0.5	8.7 ± 0.6	8.5 ± 0.7	8.5 ± 1.0	8.3 ± 0.7	8.3 ± 1.3
	125	9.7 ± 0.3	9.4 ± 0.6	9.1 ± 0.8	8.7 ± 0.6	8.6 ± 0.6	8.8 ± 0.9	8.5 ± 0.6	9.3 ± 1.0	8.6 ± 0.6	8.1 ± 0.6	8.4 ± 0.8	8.3 ± 0.7
	150	9.3 ± 0.5	9.3 ± 0.6	9.2 ± 1.0	9.0 ± 0.6	8.7 ± 0.9	8.6 ± 0.4	9.0 ± 1.2	8.7 ± 0.6	8.5 ± 0.6	8.5 ± 1.0	8.0 ± 0.7	8.2 ± 0.9
	200	9.5 ± 0.8	8.8 ± 0.6	8.8 ± 0.5	9.1 ± 0.6	8.9 ± 0.9	8.6 ± 0.6	8.3 ± 0.7	8.6 ± 0.5	8.2 ± 0.6	8.4 ± 0.9	8.1 ± 0.3	9.1 ± 1.6
	250	9.0 ± 0.4	9.0 ± 0.5	8.9 ± 0.6	8.6 ± 0.6	8.6 ± 0.7	8.6 ± 0.5	8.4 ± 0.3	8.4 ± 0.8	8.4 ± 0.5	8.2 ± 0.7	8.3 ± 0.5	8.7 ± 1.4
	300	9.1 ± 0.4	9.2 ± 0.4	8.7 ± 0.6	9.2 ± 1.1	8.4 ± 0.8	8.8 ± 0.6	8.5 ± 0.5	8.7 ± 1.4	8.6 ± 0.6	8.5 ± 1.1	9.0 ± 1.1	8.1 ± 0.5
	400	9.0 ± 0.4	9.1 ± 0.6	8.6 ± 0.7	8.2 ± 0.4	9.2 ± 1.3	8.9 ± 1.0	8.7 ± 0.9	8.6 ± 0.5	8.2 ± 0.4	8.4 ± 0.5	8.6 ± 1.1	7.9 ± 0.9
	500	9.2 ± 0.6	9.0 ± 0.6	9.0 ± 0.8	8.9 ± 0.7	8.1 ± 0.9	9.2 ± 1.2	9.0 ± 0.7	8.6 ± 0.8	8.5 ± 1.0	8.4 ± 0.9	8.0 ± 1.0	8.7 ± 1.1

B		Burnup - Number of neurons in the second hidden layer											
		10	25	50	75	100	125	150	200	250	300	400	500
Number of neuron in the first hidden layer	10	11.4 ± 1.0	10.6 ± 0.8	10.6 ± 0.8	10.3 ± 0.8	10.4 ± 1.2	10.0 ± 0.7	9.9 ± 0.7	9.5 ± 0.7	9.4 ± 1.0	9.5 ± 0.8	8.9 ± 0.7	9.1 ± 0.5
	25	10.5 ± 1.1	10.0 ± 0.8	9.4 ± 0.9	9.2 ± 0.7	9.2 ± 0.8	8.6 ± 0.8	8.7 ± 1.1	8.5 ± 0.7	8.3 ± 0.4	8.2 ± 0.6	8.7 ± 0.7	7.8 ± 1.0
	50	9.5 ± 0.5	9.5 ± 0.8	8.8 ± 0.6	8.6 ± 0.7	8.3 ± 0.6	8.4 ± 0.4	8.3 ± 0.7	8.6 ± 0.7	7.8 ± 0.6	7.9 ± 0.6	8.4 ± 0.6	7.8 ± 0.6
	75	9.1 ± 0.6	9.2 ± 0.9	8.6 ± 0.8	8.4 ± 0.6	8.6 ± 0.6	8.3 ± 0.8	8.0 ± 0.4	8.1 ± 0.6	7.9 ± 0.6	8.2 ± 0.5	7.8 ± 0.9	7.7 ± 0.8
	100	9.2 ± 0.7	9.1 ± 0.6	8.9 ± 1.1	8.6 ± 0.8	8.2 ± 0.6	8.2 ± 0.6	8.1 ± 0.9	8.1 ± 0.7	8.0 ± 0.6	7.5 ± 0.8	7.6 ± 0.9	7.6 ± 0.7
	125	9.4 ± 0.8	8.7 ± 0.9	8.8 ± 0.8	8.5 ± 0.7	8.1 ± 0.9	8.0 ± 0.5	8.3 ± 0.6	7.9 ± 0.9	8.0 ± 0.7	7.8 ± 0.9	7.9 ± 0.6	7.8 ± 0.7
	150	9.6 ± 0.9	9.4 ± 0.8	8.6 ± 0.6	8.4 ± 0.5	8.2 ± 0.6	8.8 ± 0.7	8.6 ± 0.7	8.3 ± 0.8	7.8 ± 0.7	7.9 ± 0.6	7.6 ± 0.9	7.8 ± 1.1
	200	9.7 ± 0.8	8.6 ± 0.5	8.3 ± 0.7	8.3 ± 0.8	9.2 ± 0.5	8.4 ± 0.8	8.2 ± 1.0	8.0 ± 0.7	8.2 ± 0.8	8.1 ± 0.9	8.1 ± 0.7	7.8 ± 1.3
	250	8.7 ± 0.7	8.7 ± 0.9	8.6 ± 0.9	8.4 ± 0.5	8.6 ± 0.8	8.2 ± 0.6	8.6 ± 0.7	8.1 ± 0.3	8.3 ± 0.9	7.4 ± 0.8	7.7 ± 0.8	7.4 ± 0.8
	300	8.8 ± 0.6	9.0 ± 0.5	8.4 ± 0.5	9.0 ± 1.2	8.5 ± 0.6	8.6 ± 0.7	8.4 ± 0.8	8.4 ± 0.8	8.2 ± 0.9	8.0 ± 0.9	7.7 ± 0.8	7.8 ± 0.9
	400	8.7 ± 0.8	8.5 ± 0.9	8.3 ± 0.8	8.2 ± 1.0	8.4 ± 0.8	8.4 ± 0.8	8.2 ± 0.8	8.3 ± 0.7	8.4 ± 0.6	8.3 ± 0.8	8.3 ± 1.0	7.3 ± 1.0
	500	9.0 ± 0.7	8.4 ± 0.4	8.3 ± 1.0	8.5 ± 0.9	8.1 ± 0.8	8.3 ± 0.7	8.2 ± 0.5	8.4 ± 0.7	8.1 ± 0.8	8.0 ± 1.0	7.6 ± 0.7	7.7 ± 1.1

C		Cooling time - Number of neurons in the second hidden layer											
		10	25	50	75	100	125	150	200	250	300	400	500
Number of neuron in the first hidden layer	10	3.9 ± 0.3	3.5 ± 0.4	3.4 ± 0.2	3.2 ± 0.1	3.1 ± 0.2	3.1 ± 0.2	3.0 ± 0.2	2.9 ± 0.3	2.8 ± 0.4	2.7 ± 0.3	2.7 ± 0.3	2.6 ± 0.3
	25	3.3 ± 0.2	3.1 ± 0.3	3.0 ± 0.3	2.8 ± 0.2	2.7 ± 0.2	2.6 ± 0.2	2.6 ± 0.2	2.7 ± 0.3	2.4 ± 0.2	2.4 ± 0.2	2.6 ± 0.3	2.4 ± 0.2
	50	2.9 ± 0.3	2.8 ± 0.3	2.6 ± 0.3	2.6 ± 0.3	2.6 ± 0.3	2.6 ± 0.3	2.5 ± 0.3	2.6 ± 0.3	2.5 ± 0.3	2.6 ± 0.2	2.4 ± 0.3	2.3 ± 0.3
	75	2.6 ± 0.3	2.6 ± 0.3	2.8 ± 0.2	2.6 ± 0.2	2.6 ± 0.3	2.5 ± 0.3	2.5 ± 0.3	2.3 ± 0.3	2.5 ± 0.2	2.4 ± 0.2	2.5 ± 0.2	2.3 ± 0.1
	100	3.1 ± 0.1	2.7 ± 0.3	2.8 ± 0.4	2.6 ± 0.3	2.3 ± 0.2	2.6 ± 0.3	2.6 ± 0.4	2.5 ± 0.3	2.4 ± 0.3	2.5 ± 0.3	2.6 ± 0.4	2.5 ± 0.3
	125	3.0 ± 0.2	2.7 ± 0.3	2.7 ± 0.5	2.5 ± 0.2	2.7 ± 0.3	2.6 ± 0.3	2.3 ± 0.2	2.6 ± 0.3	2.3 ± 0.2	2.6 ± 0.4	2.4 ± 0.3	2.6 ± 0.2
	150	2.9 ± 0.3	3.0 ± 0.2	2.8 ± 0.4	2.7 ± 0.3	2.7 ± 0.5	2.6 ± 0.2	2.6 ± 0.3	2.6 ± 0.3	2.5 ± 0.3	2.6 ± 0.3	2.5 ± 0.4	2.5 ± 0.4
	200	2.6 ± 0.2	2.8 ± 0.3	2.6 ± 0.3	2.9 ± 0.3	2.9 ± 0.3	2.7 ± 0.3	2.7 ± 0.2	2.7 ± 0.3	2.6 ± 0.4	2.7 ± 0.4	2.5 ± 0.4	2.5 ± 0.1
	250	2.8 ± 0.3	2.6 ± 0.2	2.8 ± 0.3	2.8 ± 0.3	2.8 ± 0.4	2.8 ± 0.4	2.6 ± 0.2	2.6 ± 0.3	2.7 ± 0.3	2.6 ± 0.3	2.5 ± 0.3	2.5 ± 0.3
	300	2.7 ± 0.2	2.8 ± 0.3	2.7 ± 0.3	2.8 ± 0.2	2.8 ± 0.4	2.6 ± 0.3	2.6 ± 0.3	2.4 ± 0.4	2.9 ± 0.3	2.6 ± 0.2	2.5 ± 0.3	2.6 ± 0.2
	400	2.9 ± 0.3	2.8 ± 0.3	2.7 ± 0.3	2.8 ± 0.2	2.6 ± 0.3	2.8 ± 0.4	2.6 ± 0.3	2.8 ± 0.5	2.7 ± 0.4	2.6 ± 0.4	2.9 ± 0.4	2.7 ± 0.5
	500	2.7 ± 0.2	2.7 ± 0.3	2.8 ± 0.3	2.8 ± 0.4	2.8 ± 0.3	2.8 ± 0.3	2.6 ± 0.2	2.5 ± 0.3	2.4 ± 0.3	2.5 ± 0.3	2.4 ± 0.3	2.4 ± 0.5

Table 2: *Mape* for the determination of the initial enrichment (top), burnup (middle), and CT' (bottom). The results for ANNs with 10, 75, and 500 neurons are highlighted for comparison.

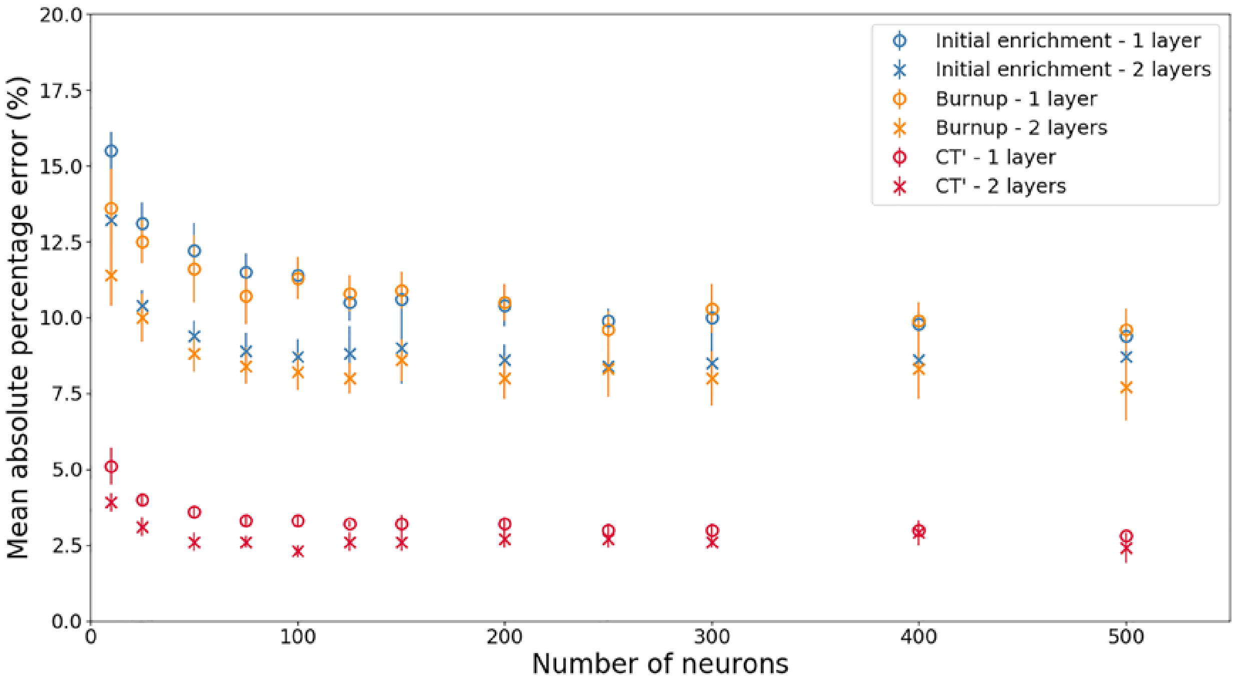

Figure 5: Mean absolute percentage error for the determination of initial enrichment, burnup, and CT'. The values refer to ANNs with one hidden layer and with two hidden layers and are shown as a function of the number of neurons in the ANN.

5. Discussion

5.1 Comparison between one hidden layer and two hidden layers ANN

The comparison of *mape* obtained for ANNs with one hidden layer and two hidden layers was conducted as the next step. All detector responses from the Forkball detector were considered as input features of the ANNs since the previous sections showed that including only a few input features led to larger *mape*. The results are shown in Figure 5 as a function of the number of neurons in the hidden layer(s). In the case of ANNs with 2 hidden layers the same number of neurons was chosen in each layer.

The results in Figure 5 show that in general the ANNs with two hidden layers reach a smaller *mape* compared to the case of ANNs with one hidden layer.

The *mape* for the initial enrichment estimate shows a decrease by increasing the number of neurons in the hidden layer from 15.5% in the case of one hidden layer ANN with 10 neurons to 9.4% for one hidden layer ANN with 500 neurons. In the case of two hidden layers ANNs the *mape* decreases from 13.2% for ANN with 10 neurons in each hidden layer to 8.7% for ANN with 500 neurons. The uncertainty associated to the *mape* due to the selection of observations in the training dataset is within 1% for almost all ANN models. A similar decreasing trend is observed for one hidden layer and two hidden layers ANNs with limited improvement of the *mape* by increasing above 100 the number of neurons in the ANNs.

The *mape* for the burnup estimate shows a decrease by increasing the number of neurons in the hidden layer from 13.6% in the case of one hidden layer ANN with 10 neurons to 9.6% for one hidden layer ANN with 500 neurons. In the case of two hidden layers ANNs the *mape* decreases from 11.4% for ANN with 10 neurons in each hidden layer to 7.7% for ANN with 500 neurons. The uncertainty associated to the *mape* due to the selection of observations in the training dataset is within 1% for almost all ANN models. A similar decreasing trend is observed for one hidden layer and two hidden layers ANNs with limited improvement of the *mape* by increasing above 100 the number of neurons in the ANN.

The *mape* for the CT' estimate shows a decrease by increasing the number of neurons in the hidden layer from 5.1% in the case of one hidden layer ANN with 10 neurons to 2.8% for one hidden layer ANN with 500 neurons. In the case of two hidden layers ANNs the *mape* decreases from 3.9% for ANN with 10 neurons in each hidden layer to 2.4% for ANN with 500 neurons. The uncertainty associated to the *mape* due to the selection of observations in the training dataset is within 0.5% for almost all ANN models. A similar decreasing trend is observed for one hidden layer and two hidden layers ANNs with limited improvement of the *mape* by increasing above 50 the number of neurons in the ANN.

The results from this study are in general agreement with earlier ANN models developed at SCK CEN [15]. Previous research concluded that ANNs were able to estimate the initial enrichment within 2% for 98% of the cases, the

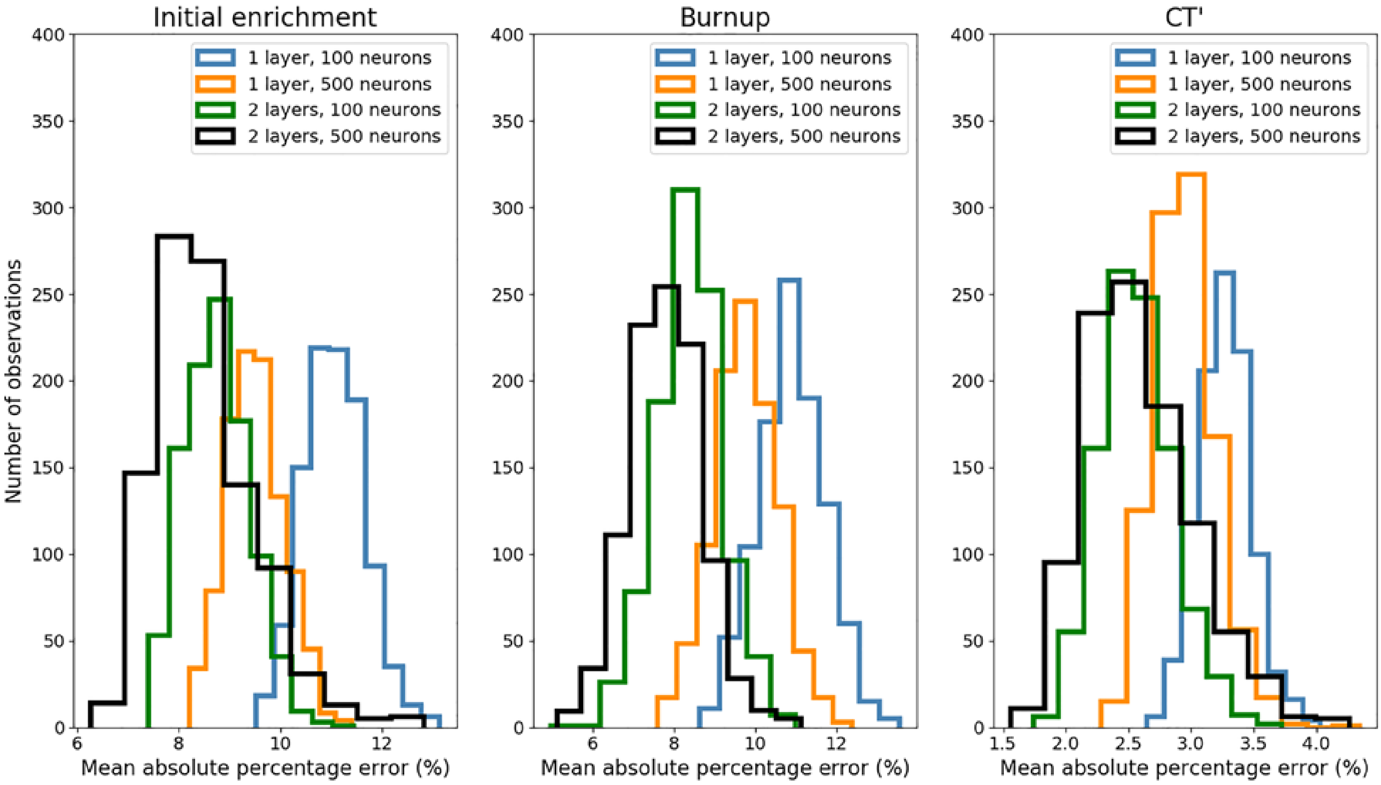

Figure 6: distribution of mean absolute percentage error for the determination of initial enrichment, burnup, and CT'. The training and testing of each ANN architecture was repeated for 1000 iterations, each time with a random partition of the dataset.

(a) Initial enrichment	1 hidden layer,	1 hidden layer,	2 hidden layers,	2 hidden layers,
Average	11.1	9.5	8.8	8.5
Standard deviation	0.6	0.6	0.6	1.0
Rel. standard deviation	5%	6%	7%	12%

(b) Burnup	1 hidden layer,	1 hidden layer,	2 hidden layers,	2 hidden layers,
Average	10.9	9.8	8.4	7.8
Standard deviation	0.8	0.8	0.8	0.9
Rel. standard deviation	8%	8%	10%	11%

500 neurons	1 hidden layer,	1 hidden layer,	2 hidden layers,	2 hidden layers,
Average	3.3	3.0	2.6	2.6
Standard deviation	0.2	0.2	0.3	0.4
Rel. standard deviation	6%	8%	11%	17%

Table 3: Mean absolute percentage error average, standard deviation, and relative standard deviation for the determination of initial enrichment (top), burnup (middle), and CT' (bottom).

burnup within 3% for 96% of the cases, and the cooling time within 10% for 87% of the cases. However, the values reported in previous research refer to estimates in the training dataset, therefore they should be considered as overestimations of ANNs accuracy. The reason for the large error for the cooling time estimate is probably due to lack of data processing, different activation function, and optimization algorithm used in previous research.

The performance of ANNs with 1 hidden layer and 100 neurons are in line also with published work on the determination of spent fuel parameters using the Fork detector. Research [33] showed that by using calibration measurements, difference between measured and declared burnup is within 2% for cooling times longer than 3 years and burnup between 30 and 55 GWd/tU. The deviation increases outside these validity ranges up to 27%. Results from verification campaigns using the Fork detector showed that the relative standard deviation between measured and calculated count rates is less than 8% for neutron detectors and less than 7% for gamma-ray detectors [34].

Data analysis procedures applied in previous work [33],[34] are based either on calibration curves or rely on operator data to estimate the spent fuel parameters. In contrast, the ANNs developed in this study reach a similar performance in the estimation of the fuel parameters without additional input features than the detector responses.

5.2 Optimization of ANN size to reduce overfitting

A set of ANNs with different architectures was selected for further comparison. ANNs with either one hidden layer or two hidden layers and either 100 neurons or 500 neurons in each layer were chosen. The ANNs with 100 neurons were chosen because in the previous section they showed small improvements in the $mape$ compared to larger ANNs. Several rule of thumbs have been proposed to link the size of the ANN to the minimum dataset size to obtain reliable estimates [35]. It is generally thought that larger ANNs require larger amount of data to converge, and smaller ANNs are in general preferred because they tend to reduce the risk of overfitting the training dataset. Therefore, the objective of this section is to optimize the ANN size in order to reduce overfitting.

For each ANN the training and testing was repeated for 1000 iterations, each time with a random partition of the dataset. The $mape$ was recorded for each iteration and the distribution is shown in Figure 6. The $mape$ average value, standard deviation, and relative standard deviation compared to the average value were calculated for each ANN architecture and are summarized in Table 3.

The distributions shown in Figure 6 follow quite well the shape of a normal distribution. However, in the case of ANNs with two hidden layers and 500 neurons in each hidden layer the distributions for the determination of initial enrichment and cooling time show a long tail on the high-$mape$ side of the distribution. This can be an effect of the overfitting of the dataset due to the large size of the ANNs.

The values included in Table 3 highlight the reduction of the $mape$ average value by increasing the size and number of hidden layers. However, the table shows also that the decrease of the $mape$ average value is countered by the increase of the $mape$ standard deviation and relative standard deviation compared to the average value. The comparison in this section indicates that ANNs with 1 hidden layer and 100 neurons are already effective in inferring initial enrichment, burnup, and CT’ of spent fuel assemblies. Further enlarging the ANN architecture leads to an increase in the relative standard deviation of the estimate and risk of model overfitting.

6. Conclusions

Several ANNs were developed using as input features the simulated detector responses of the Forkball detector with the aim of inferring the initial enrichment, burnup, or cooling time of spent fuel assemblies. ANN models with one hidden layer and two hidden layers were considered, setting the number of neurons as hyper-parameter in the study. The ANNs performance was measured with the $mape$ between the predicted and declared value of the output feature.

The results from ANNs with one hidden layer showed that combining all detector responses from the Forkball detector leads to a decrease of the $mape$ compared to the cases using only one detector response. In general it was observed that the $mape$ decreases by increasing the number of neurons in the hidden layer, but the reduction is larger up to 125 neurons and the $mape$ remains rather stable by further increasing the number of neurons. The data processing of the cooling time variable was essential to obtain a reliable estimate from the ANN. The CT’ feature, obtained with a logarithmic function from the cooling time, was used throughout the study to obtain an estimate of the cooling time because ANNs using the cooling time feature obtained very large $mape$. The ANNs with 500 neurons in the hidden layer were able to estimate the initial enrichment with a $mape$ of 9.4%, the burnup with a $mape$ of 9.6%, and the cooling time - via the CT’ feature - with a $mape$ of 2.8%. The uncertainty associated to the $mape$ due to the selection of observations in the training dataset is within 1% for almost all cases.

Considering the results from the ANNs with one hidden layer, ANNs with two hidden layers were developed only using all features from the Forkball detector, and processing the cooling time variable for the corresponding ANNs. The results from ANNs with two hidden layers showed a reduction of the $mape$ by increasing the number of neurons in the hidden layers, but the decrease is rather limited for ANNs with more than 75 neurons. It was observed also that the $mape$ is slightly smaller for ANNs with equal number of neurons in each hidden layer. The ANNs with 500 neurons in both hidden layers were able to estimate the initial enrichment with a $mape$ of 8.7%, the burnup with a $mape$ of 7.7%, and the cooling time – via the CT’ feature - with a $mape$ of 2.4%. The uncertainty associated to the $mape$ due to the selection of observations in the training dataset is within 1% for the estimates of initial enrichment and burnup, and within 0.5% for the estimate of cooling time.

The $mape$ average value decreases by increasing the number of neurons and the number of hidden layers in the ANNs. However, this effect is countered by the increase of the $mape$ standard deviation and relative standard deviation compared to the $mape$ average value.

Based on the results presented in the paper, and given the size of the available dataset, it is recommended to use ANNs with 1 hidden layer and 100 neurons for the estimation of the spent fuel parameters. Such ANNs are already effective in inferring the initial enrichment and burnup within 12%, and the cooling time – via the CT' feature - within 4%. The deviation between declared values and estimates from the ANNs are similar to data analysis procedures used for the Fork detector. However, current data analysis procedures rely either on calibration curves or on operator data, whereas the ANNs developed in this study require only the detector responses as input features.

Future work will focus on the optimal ANN configurations obtained in this study to evaluate if the $mape$ of the developed ANNs are constant over the range of initial enrichment, burnup, and cooling time. The possibility of simultaneous estimation of the three output parameters by a single ANN will also be investigated.

7. Legal matters

7.1 Privacy regulations and protection of personal data

I agree that ESARDA may print my name/contact data/photograph/article in the ESARDA Bulletin/Symposium proceedings or any other ESARDA publications and when necessary for any other purposes connected with ESARDA activities.

7.2 Copyright

The authors agree that submission of an article automatically authorises ESARDA to publish the work/article in whole or in part in all ESARDA publications – the bulletin, meeting proceedings, and on the website.

Once the article has been accepted for publication the copyright is reserved, but part of the publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopy, recording, or otherwise, provided that the source is properly acknowledged.

The authors declare that their work/article is original and not a violation or infringement of any existing copyright.

8. References

[1] International Atomic Energy Agency (IAEA), 1970. Treaty on the non-proliferation of nuclear weapons. INFCIRC/140.

[2] International Atomic Energy Agency (IAEA), 2020. Infographic: safeguards implementation 2020. Available at: https://www.iaea.org/sites/default/files/21/06/sg-implementation-2020.pdf. Last accessed: 13/09/2021.

[3] International Atomic Energy Agency (IAEA), 2011. Safeguards techniques and equipment, 2011 edition" International nuclear verification series no. 1 (rev. 2).

[4] Borella A., et al., 2014. Sensitivity studies on the neutron emission of spent nuclear fuel by means of the ORIGEN-ARP code. Proceedings of the 2014 INMM annual meeting.

[5] Borella A., et al., 2019. Determination of 239Pu content in spent fuel with the SINRD technique by using artificial and natural neural networks. ESARDA Bulletin n.58.

[6] Rossa R., et al., 2020. Comparison of machine learning models for the detection of partial defects in spent nuclear fuel. Annals of Nuclear Energy 147 (2020) 107680.

[7] Elter Z., et al, 2020. A methodology to identify partial defects in spent nuclear fuel using gamma spectroscopy data. ESARDA Bulletin n.61.

[8] Bachmann A. M., et al., 2021. Comparison and uncertainty of multivariate modelling techniques to characterize used nuclear fuel. Nuclear Instruments and Methods for Physics Research A 991 (2021) 164994.

[9] Warner, T., et al., 2018. Exploitation of high-frequency acquisition of imagery from satellite constellations within a semi-automated change detection framework for IAEA safeguards purposes. IAEA Symposium on International Safeguards, 5-8 November 2018.

[10] Cui, Y., et al., 2018. Using deep machine learning to conduct object-based identification and motion detection on safeguards video surveillance. IAEA Symposium on International Safeguards, 5-8 November 2018.

[11] Feldman, Y., et al., 2018. Toward a multimodal-deep learning retrieval system for monitoring nuclear Proliferation Activities. Journal of Nuclear Materials Management, Vol. 46, No. 3 (2018).

[12] Gastelum, Z., et al., 2018. Inferring the operational status of nuclear facilities with convolutional neural networks to support international safeguards verification. Journal of Nuclear Materials Management, Vol. 46, No. 3 (2018).

[13] Sánchez-Belenguer C., et al., 2020. RISE: A Novel Indoor Visual Place Recogniser. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

[14] Lin Y., et al., 2021. Using machine learning to track objects across cameras. Proceedings of the INMM & ESARDA Joint Virtual Annual Meeting.

[15] Borella A., et al., 2017. Signatures from the spent fuel simulations and interpretation of the data with neural network analysis. ESARDA Bulletin n.55.

[16] Dim O., et al., 2021. Verification of Triso fuel burnup using machine learning algorithms. Proceedings of the INMM & ESARDA Joint Virtual Annual Meeting.

[17] Borella A., et al., 2014. Advances in the development of a spent fuel measurement device in Belgian nuclear power plants. In: Proceedings of the 2014 IAEA Symposium on International Safeguards – Linking Strategy, Implementation and People.

[18] Pelowitz, D., editor, 2011. MCNPX user's manual version 2.7.0. Los Alamos National Laboratory LA-CP-11-00438.

[19] Leshno M. et al., 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural networks, 6(6):861-867

[20] Choy C. B., et al., 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. European conference on computer vision – ECCV 2016.

[21] Schechner S., 2017. Facebook Boosts A.I. to Block Terrorist Propaganda. Wall Street Journal. https://www.wsj.com/articles/facebook-boosts-a-i-to-block-terrorist-propaganda-1497546000. Last accessed 13 September 2021.

[22] Nix R., et al., 2017. Classification of Android apps and malware using deep neural networks. 2017 International Joint Conference on Neural Networks (IJCNN): 1871–1878.

[23] Kamuda M., et al., 2021. Automated Isotope Identification Algorithm Using Artificial Neural Networks. DOI 10.1109/TNS.2017.2693152, IEEE Transactions on Nuclear Science.

[24] Hyun Lee J., et al., 2020. An online operator support tool for severe accident management in nuclear power plants using dynamic event trees and deep learning. Annals of Nuclear Energy Volume 146, October 2020, 107626.

[25] Baxt W. G., 1995. Application of artificial neural networks to clinical medicine. The Lancet Volume 346, Issue 8983, 28 October 1995, Pages 1135-1138.

[26] Baijayanta R., 2020. All about Feature Scaling. https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35. Last accessed on 13/09/2021.

[27] https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b. Last accessed 14/11/2021

[28] Ciresan D., et al., 2011. Flexible, High Performance Convolutional Neural Networks for Image Classification. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two. 2: 1237–1242.

[29] https://scikit-learn.org/stable/modules/cross_validation.html. Last accessed on 13/09/2021.

[30] Goodfellow I., et al., 2016. Deep Learning (Adaptive Computation And Machine Learning Series). The MIT Press.

[31] Kingma D. P., et al., 2015. Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference for Learning Representations.

[32] https://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-percentage-error. Last accessed on 10/11/2021.

[33] Borella A., et al., 2011. Spent Fuel Measurements with the Fork Detector at the Nuclear Power Plant of Doel. Proceedings of the 33rd ESARDA annual meeting.

[34] Vaccaro S., et al., 2018. Advancing the Fork detector for quantitative spent nuclear fuel verification. Nuclear Instruments and Methods in Physics Research, A 888 (2018) 202–217.

[35] Alwosheel A., et al., 2018. Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. Journal of Choice Modelling 28 (2018) 167-182.