EPJ Web of Conferences **302**, 17011 (2024) https://doi.org/10.1051/epjconf/202430217011
*SNA + MC 2024*

# Exploration of genetic algorithms to build a balanced neutron spectra dataset useful to train unfolding techniques based on artificial neural networks

*Maha Bouhadida[1]*, Rodayna Hmede[2], Mariya Brovchenko[1], Wilfried Monange[1], Thibaut Vinchon[2], Quentin Ducasse[2], and François Trompier[1]*
*[1] Institut de Radioprotection et de Sûreté Nucléaire (IRSN), Fontenay-aux-Roses 92260, France[2] Institut de Radioprotection et de Sûreté Nucléaire (IRSN), Cadarache 13115, France*

**Abstract.** Diverse domains need neutrons unfolding technics to assess the incident neutron energy spectrum. Examples are radiation protection, nuclear reactor physics or criticality safety. Traditionally, methods based on the Bayesian approach requires an initial guess of the solution which may significantly impact the unfolding result. This work proposes a novel method for neutron spectrum reconstruction using machine learning (ML) techniques trained on a large dataset. To ensure the ML algorithm to perform on a large domain of application particular attention has been paid to the dataset creation. We propose a comparison of two methods of building large dataset where the most adequate solution is obtained using a dynamic genetic algorithm (GA). This GA targets optimal combinations of 48 parameters to generate a variety of neutron spectra. The resulting dataset is then used to train a new convolutional neural network architecture for unfolding neutron spectra. Obtained performance metrics of the tested architecture show high efficiency and emphasize the added value of the built dataset.

## 1 Introduction

Neutron spectrum assessment has always been a very hard task. To the best of our knowledge, there are currently no experimental devices for direct measurement of the neutron spectrum on a wide energy range of neutrons (typically from eV to MeV) [1]. The conventional measurements approaches are the so-called Bonner Spheres Spectroscopy (BSS) and neutron activation foils.

For the BSS approach, a detector - mostly sensitive to thermal neutron - is inserted at the centre of a polyethylene spheres. Multiple spheres with different diameter allow to get different response function.[2]. Generally, from 8 to 12 spheres are used. The response function of detector can be determined experimentally or by simulation. Using these response functions and a few channels unfolding algorithm, the neutron spectrum can be obtained from the knowledge of the vector constituted in the counting rates obtained with the different spheres. It is considered as an inverse physical problem.

Activation foil techniques use materials transmuted to one or several radioisotopes after neutron capture. Multiple material allow to get different response function. The input data are then the activities of the different radioisotopes in the different materials induced by neutron interaction.

* Corresponding author: maha.bouhadida@gmail.com

Computational programs classically used to unfold the spectra from the measurements are based on iteration, maximal entropy, matrix inversion, etc. [3]. However, several limitations have been reported such as the requirement of prior information like a prior neutron spectrum [4].

Consequently, unfolding techniques relying on artificial neural networks (ANNs) have become a subject of interest [4]. We may mention many developed unfolding ANN models like multi-layer perceptron (MLP) architecture [5], Radial function basis networks [6], generalized regression neural networks, etc. [7]. ANNs may allow to overcome the prior information constraint at the cost of a large training dataset containing physical neutron spectra. According to the recent literature, this aspect is insufficiently addressed since the used training datasets are either real system data containing a limited number of neutron spectra or numerical data containing mathematical shapes (Maxwellian, Gaussian, 1/watt…) [8]. It is worth mentioning here, that the authors have not had access to a sufficient large experimental measurement dataset. Moreover, as the neutron spectra cannot be directly measured experimentally, the dataset must rely at least partially on the neutronic simulations or numerical data. Besides, the training dataset should not only be large but also balanced. In other terms, it should allow the neural network to predict with equivalent precision and performances for all the considered energy range. To make thinks easer the three following energy ranges have been considered: thermal E < 1eV, epithermal: 1eV < E < 10 keV and fast E > 10 keV. To target an equivalent level of precision within those domains, we must ensure that they roughly have the same quantity of data available within each energy range.

In this paper, we explore different dataset-building techniques allowing the generation of as many as needed physical neutron spectra and in the described balanced way. We explore genetic algorithms (GA) [9] to find optimal physical contexts leading to the cited objective. These evolutionary algorithms are inspired by Darwin's population evolving theory and they are known for their effective adaptability to different problems dealing with the search for optimal solutions or combinations. The building process is detailed, and the generated dataset is associated with reaction rates over a detector and fed to a recently deployed convolutional neural network (CNN) [10][11]. Performances are discussed to emphasize the added value of smart technology in the nuclear field.

## 2 Physical model

The building concept of the dataset has been briefly discussed in the following references [10,11]. This dataset includes neutron spectra and the corresponding reaction rates within the detector. For the following simulations, the considered detector is the multiple foil neutron activation spectrometer (SNAC2) described in reference [12]. SNAC2 is composed of foils of Cu (x3), Au, Ni, Mg, In (x2), and ebonite with Sulphur. It is used as a critical accident area neutron spectrometer in French Nuclear combustible processing plants and for reactor dosimetry.

Fig. 1 details the dataset conception steps, and the link between spectra and reaction rates. The neutron fluence at each energy bins are the outputs of the ANN and the associated reaction rates are its inputs. The reaction rate set is found by applying Equation 1 to the generated neutron spectra set using the response functions of SNAC2. In Equation 1, $\varphi_i$ represents the flux at the ith energy bin, $RF_{ji}$ is the response function value of the jth activation foil at the ith energy bin and $R_j$ is the jth reaction rate (j :1 to 8), and n is the total number of bins. The response functions $RF_{ji}$ are calculated using Monte Carlo simulations by modelling the SNAC2.

$$R_j = \sum_{i=1}^{n} RF_{ji} \times \phi_i \quad (1)$$

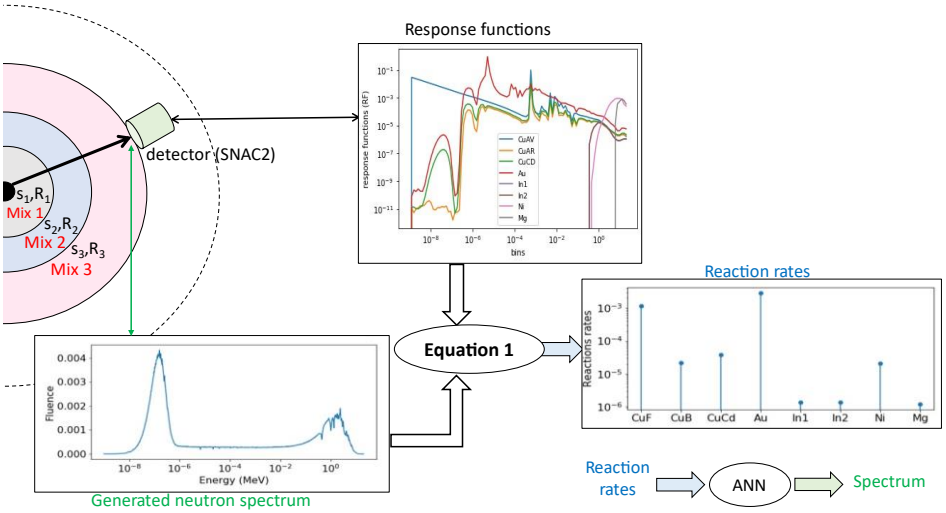



**Fig. 1.** The building steps of the dataset.

The set of neutron spectra aims to cover a large variety of physical fission nuclear systems and to be balanced. The spectra creation is based on Monte Carlo simulation, that are built with a simple physical model (neutron source, material's geometry, and detector) using the SERPENT 2 [12] software. The described technique aims to generate as many spectra as possible, while generating realistic spectra shapes through sampling the input parameters. The simulated system is made of three concentric crowns of material mixture to mimic heterogeneous systems. These mixtures are made of materials commonly encountered in nuclear facilities and having an effect on neutron energy. Material can be categorised as: 1-moderators, that slow down the neutrons and tends to create a more thermal spectrum, 2-neutron absorber, that causes neutrons disappearance in some peculiar energy domains.

The designed geometry is composed of a central sphere s1 of a radius R1 surrounded by two spherical "shells" s2 and s3 of respective radii R2 and R3. The neutron source is located at the center of s1 and $3x10^7$ neutrons are emitted passing through the three cells (s1, s2 and s3) to generate spectra. The outgoing neutron flux is detected at the external surface of s3 and stored over 1000 energy bins between $10^{-11}$ MeV and 20 MeV. The first bin covers the energy interval [$10^{-11}$ MeV, $10^{-9}$ MeV], and others have equal lethargy between $10^{-9}$ MeV and 20 MeV. The generated default SERPENT spectra are then normalized by the total neutron flux. The radii R1, R2 and R3 are calculated depending on randomly chosen thicknesses e0, e1 and e2 in which R1 = e0, R2 = R1 + e1 and R3 = R2 + e2.

Neutron are created from fission sources randomly chosen among the following isotope list: U-235, U-238, Pu-239, Pu-241. Along the neutrons' path, each cell (s1, s2 and s3) includes a mixture of materials that absorb and/or slow down neutrons. The selected materials are listed in Table 1.

*Table 1: Materials used in the simulations.*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Graphite | concrete | steel 304 | $B_4C$ | $UO_2$ | U8 | Pb | Ag/In/Cd mixture |
| Gadolinium | Hafnium | Xenon | $CH_2$ | $D_2O$ | $H_2O$ | Air | Vacuum |

Note that vacuum is added to the list to simulate variation of the material densities. Thus, there are 16 materials (15 chemical elements and vacuum) at each cell with a fraction sum equal to unity. The description of moderator materials ($H_2O$, $D_2O$, graphite, etc.) includes the S(α,β) that corrects the elements cross sections by taking into account chemical bonds and is rather sensitive to the temperature of the medium. The temperature of S(α,β) is randomly chosen per simulation in the set [21°C, 51°C, 101°C, 151°C, 201°C, 251°C, 300°C, 350°C, 373°C, 523°C, 726°C]. As a result, 53 parameters have to be randomly chosen (R1, R2, R3, isotope of the neutron source, temperature of S(α,β) and 48 material fractions, i.e 16 in each cell) to generate a set of different systems that produces variety of neutron spectra at the external surface of the second shell s3.

# 3 Dataset generation methods

## *3.1* Probability distribution-based method

Our goal being to find out a set of neutron spectra that represents all the systems that we may encounter in the fission nuclear systems, we initially sample the 53 parameters using various distributions to ensure a homogeneous coverage of the input parameter space. We used equiprobable distributions for source selection and for the S(α,β) temperature for all the materials in each simulation. The sampling of the mass fractions in the three materials is less trivial, because of the complex non-linear neutron interaction inside matter. These fractions are chosen randomly following a continuous probability distribution, such as the exponential or beta distributions.

Despite additions of additional constraints, such as imposing high fractions for moderator materials, or a full vacuum cell, etc..., this approach has not helped to build a balanced set of generated spectra as illustrated in the results section. We then decided to rely on smarter algorithms and choose genetic algorithms.

### *3.2* GA based solution

Darwin's principle of "Survival of the fittest" can be used to introduce the genetic algorithm principle as the entire population of the ecosystem evolves to fit the new environmental requirements. GA, which is inspired by this theory, aims to search for optimal solutions to a problem [13,14,15]. What makes GA efficient is the simplicity of its approach, its flexibility, and its robust adaption to the circumstances. Besides, it can be applied to non-linear problems defined on discrete, continuous, or mixed search spaces. However, the challenge of GA is the appropriate choice of the objective function that should generally be minimized and of the fitness coefficient which is generally depending on the former.

GA has been used to find the optimal fractions set that enable the conception of a balanced neutron spectra set used to train ANNs. All other input parameters (R1, R2, R3, isotope of the neutron source, temperature of S(α,β) ) are chosen randomly and fixed before performing GA to find the suitable fractions.

GA individuals are vectors of 48 fractions (16 fractions per cell). The objective function and the fitness coefficient play a significant role for GA. They are depending on a key parameter (so called ”Ratio”) and is described through Equation 2. φ(x) represents the neutron flux as a function of the energy (“E” in MeV) for an individual (a set of 48 fractions). The Ratio parameter represents the contribution of the "thermal" part to the neutron spectrum. Thus, a spectrum with Ratio close to 1 is labelled as thermal and the one with Ratio close to 0 is considered as fast. For the middle values of Ratio, the spectrum is considered as equilibrated.

$$\text{Ratio} = \frac{\int_{10^{-9}}^{10^{-6}} \varphi(\text{E})\,\text{dE}}{\int_{10^{-9}}^{10^{-6}} \varphi(\text{E})\text{dE} + \int_{10^{-2}}^{10^{2}} \varphi(\text{E})\text{dE}} \quad (2)$$

The diagram presented in Fig. 2(a) details the whole algorithm including the so called dynamically genetic part (dyn_GA), that we have implemented. The concept is to build a balanced set of neutron spectra by iteratively targeting the less populated ratios. We consider that the ratio distribution is optimized when all the ratios from 0 to 1 have almost equivalent redundancies. This means that the associated spectra set is made up of the different spectrum types with equal portions as explained above.

The neutron spectra and the associated ratios are updated after each generation by dyn_GA that is presented in Fig. 3(b). The ratio redundancy histogram is established each time dyn_GA stops running. The established histogram allows the dynamic detection of a new target ratio rt whose redundancy is the lowest. It can be described, based on Equation (2), at each dyn_GA iteration, by the following expression (Equation 3):

$$r_t = Ratio,\ with\ lowest\ redundancy \quad (3)$$

The target ratio $r_t$ is then the key parameter that will be explored for the following dyn_GA call. It is dynamically updated after each ratio redundancy computation. Based on its value, the objective function and the fitness coefficient are updated too since both depend on it. The whole algorithm stops when the ratio redundancy distribution becomes optimized. To limit the computation time, we assume that this condition is reached when the relative gap ($g_r$) is under 15% that is defined by Equation (4), where $r_{mean}$ is the mean of Ratio:

$$g_r = \frac{|r_{mean} - r_t|}{r_{mean}} \quad (4)$$

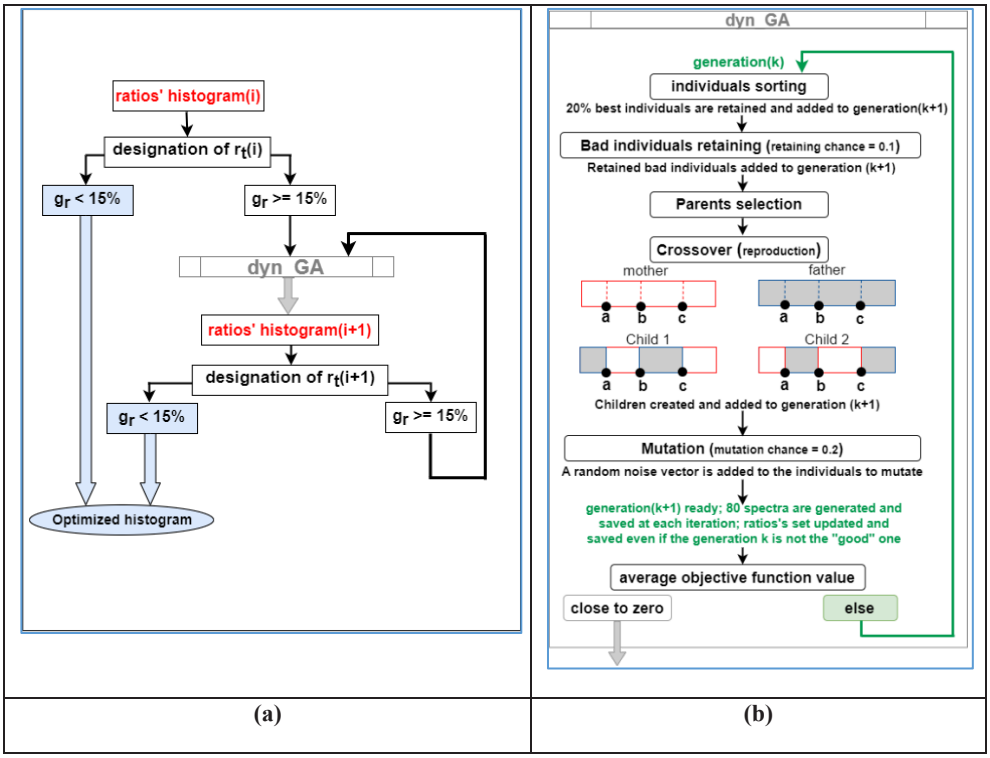



**Fig. 2.** (a): scheme of the whole algorithm; (b) detailed scheme of dyn_GA

The first step of the whole algorithm is to define the initial population. This initial population contains 80 individuals (80 vectors of 48 fractions). For each individual, every 16 consecutive fractions represent a cell in the SERPENT geometry (s1, s2 or s3) and have a sum of 1. This special structure of the vector is taken into consideration in the next steps, especially for the normalization process. The ratios of the initial population are calculated, and their redundancy distribution is established. Then, the target ratio rt is fixed and $g_r$ is calculated. If $g_r$ is higher than 15%, dyn_GA re-starts running.

Fig.3(b) details dyn_GA steps. The initial population is evolved through several generations until finding an average objective function close to zero which is the stopping condition of dyn_GA. The first step of the evolving process is to sort the initial individuals according to their fitness coefficients. In fact, the objective function $f_{obj}$ defined by Equation (5) is computed for each individual leading to the fitness coefficients (Equation 6). Since $r_t$ is updated after each dyn_GA iteration k, the value of the next objective function ($f_{obj}$) and the fitness coefficients are subsequently updated, at iteration k+1, by considering this new rt value.

$$f_{obj}\,(\text{individual}) = \text{Ratio(Individual)} - r_t \tag{5}$$

$$\text{fitness}_{coef}(\text{individual}) = \frac{f_{obj}\,(\text{individual})}{\sum_{i=1}^{80} f_{obj}\,(i)} \tag{6}$$

The best 20% of individuals in terms of fitness of each population are first retained for the next generated one. To ensure the diversification of the new population and to better explore input parametric spaces, we decided to retain some bad individuals in terms of fitness outside of the best 20% with a probability of 0.1. Then, the parents are randomly designated

among the total retained individuals. This couple allows the creation of children by crossover. The considered crossover technique is modelled in Fig.3(b). In fact, 3 random positions a, b and c are chosen at each crossover iteration and two children are created. These two children individuals are a linear combination of the parents using a, b and c. The crossover loop turns until the maximal number of the generation individuals is reached (80 in our case). Each created child is normalized: each consecutive 16 elements (fractions) are normalized by their sum. This technique already explained above allows the generation of an appropriate fraction set. The new population in the first generation is then composed of the 20% best individuals of the previous generation, some "bad" individuals and children created from chosen parents. The final step of evolution is mutation. The probability of mutating an individual is set to 0.2. to add random noise to some individuals of the new population. The same normalization technique is also applied at this step. Since the first generation is created, the neutron spectra associated with the fractions set are computed following the detailed description above. The spectra set allows the calculation of the first generation's ratios and the population average objective function is computed. If it is close to zero, dyn_GA stops running, and the new ratios' redundancy distribution is established. Otherwise, the evolving process is repeated through many generations until reaching an average objective function close to zero. At each generation, the neutron spectra set and the ratio set are updated. When the average of the objective functions has reached a value below a given threshold, the ratio redundancy distribution is automatically updated, the new $r_t$ is dynamically found and the new $g_r$ is calculated to decide if the whole algorithm should continue running. The described steps are repeated until an optimized redundancy distribution is obtained.

## 4 Artificial neural networks

The generated dataset is used to train a neural network based on transpose convolutional blocks called “ResConvT” and recently detailed in reference [10]. The input, which is the reaction rates set, is feeding a multilayer perceptron network (MLP) [16]. The output of MLP represents the input of the first ResConvT block (see Fig. 3). After each ResConvT block, the data dimension is doubled and after the fourth block, a dense layer gives a final output with a dimension of 1000 elements (i.e. energy bins). ResConvT is composed of two branches where the first is an upsampling branch and the second is a transpose convolution operation [17]. Both branches’ results are summed and sent to the activation function. The parameters used of the branches’ layers are specified in reference [10]. details of the general architecture of the used CNN are illustrated in Fig.3.

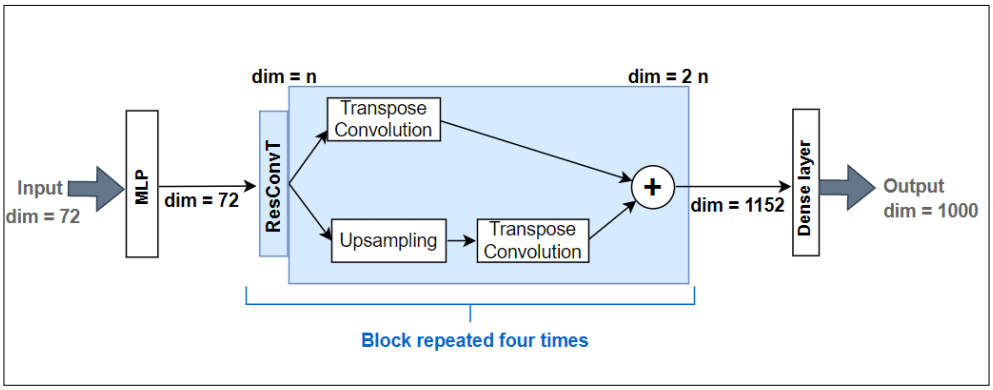



**Fig. 3.**Scheme of the explored CNN architecture (ResConvT [10]); n is the input dimension before each ResConvT block and 2n is its output

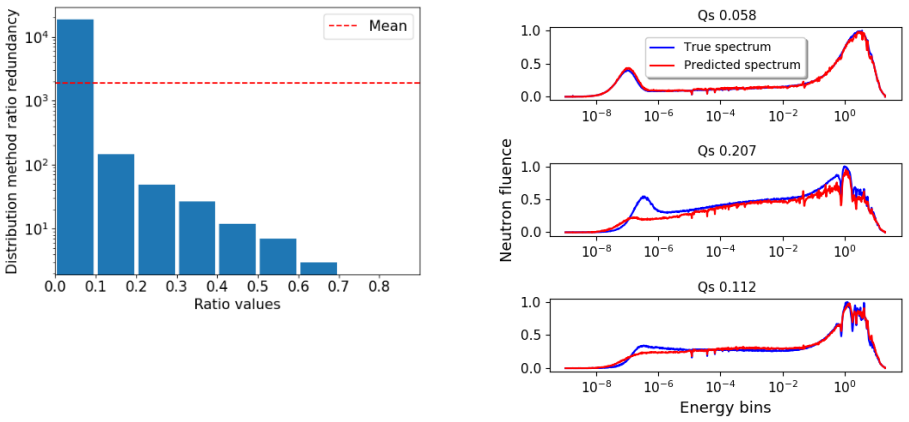
The processing of the spectra and reaction rates is a mandatory step before training an ANN. The aim is to find the processing combination leading to the best performance of the deployed ANN. Feature engineering is performed by computing the ratios between all the reaction rates pairs and by normalizing the rates by their sum. Then we propose two possible ANN input shapes: (number of samples, 64 = 8 rates + 56 computed ratios) or (number of samples, 72 = 8 rates + 56 computed ratios + 8 normalized rates). Various standard scaling methods have been applied to both spectra and reaction rates. For each fixed spectrum processing, three reaction rates' processes ("MaxScaler", "MinMaxScaler", "StandardScaler") as defined in the documentation [18] are tested. The tests are respectively done for reaction rates' vectors of 64 values and 72 values. We decide to consider, for the simulations detailed above, the MinMaxScaler over bins for the spectra set and the StandardScaler for reaction rates with a dimension of 72 for the next results. We considered 75% of the dataset for training and the other 25% for validation in both methods. The training process has been performed using an Nvidia Tesla V100 GPU (12GB of GPU memory) and the running time is about one hour for the described case. The considered training performance metric for the following tests is the spectral quality ($Q_s$), whose formula is detailed in Equation 7 (the sums is computed over the energy bins). $Q_s$ represents the closeness of the predicted neutron spectrum $y_{pred}$ to the true one $y_{true}$ . The lower the Qs value, the better the agreement between the predicted and true spectrum is.

$$Q_s = \sqrt{\frac{\sum_1^n (y_{pred} - y_{true})^2}{\sum_1^n (y_{true})^2}} \tag{7}$$

## 5.Results and discussion

### *5.1 Training with the dataset generated by the probability distribution method*

Fig.4.a represents the redundancy distribution of the dataset generated using the probability distribution methods. The resulting dataset is mainly composed of fast spectra (almost 99%). Fig.4.b presents three examples of its spectra and their reconstruction by the ANN. As shown by this figure and the associated $Q_s$ values, the predictions show inaccurate reconstruction of key structural part of the neutron spectra, hence the need of a more balanced dataset for the training of the ANN.

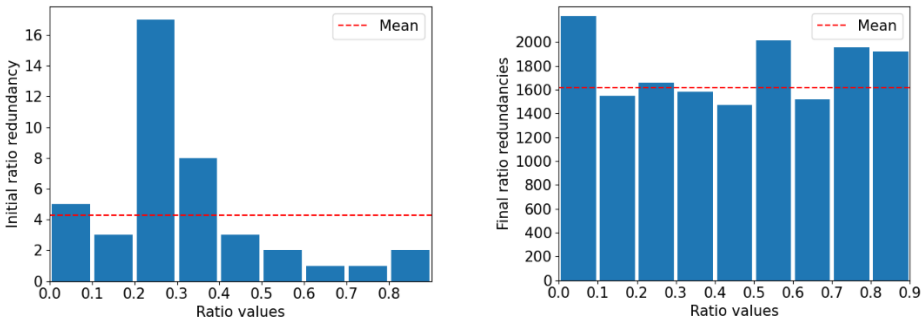



(a) (b)

**Fig. 4.** (a) Ratio redundancy distribution (in y-log scale) of the dataset – (b) Examples of spectrum reconstructions using ResConvT (probability distribution method)

### *5.2 Training with the dataset built by dyn_GA*

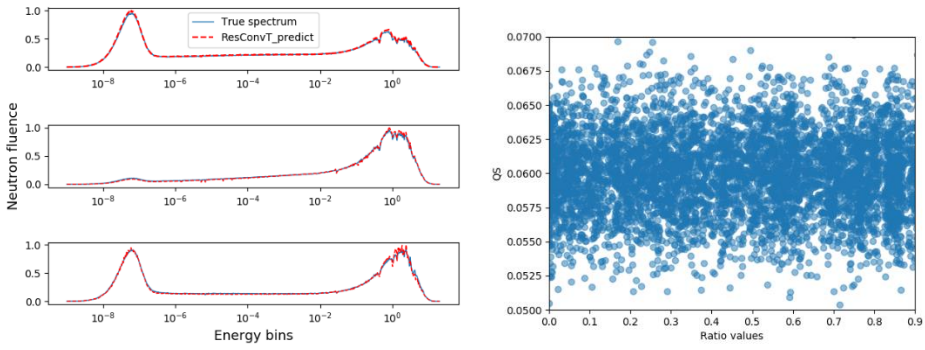
Fig.5 (a) shows the ratio distribution of the considered initial population obtained by an initial sampling in the unbalanced population issued from the previous approach. Since in our algorithm, the new $r_t$ is defined as the first minimum bin (in terms of redundancy) found in the histogram, the initial $r_t$ is computed at 0.65 and $g_r$ is about 80%. Fig. 5(b) shows the final ratios distribution at the end of calculation. The final $g_r$ value drops to 14.6%. 18 $r_t$ were sequentially dynamically chosen and 202 generations of 80 individuals have been generated leading to 16 240 (80 + 202 x 80) neutron spectra. The space of ratio values is well explored and the proportion of thermal, fast, and epithermal neutron spectra are respectively 36%, 35% and 29%. This result can be improved by decreasing the objective value $g_r$, tunning the probabilities and the population count per iteration, but it was found as shown below to be sufficient for training our ANN architecture.




(a) (b)

**Fig. 5.** Ratio redundancy distributions: (a) of the initial population; (b) of the final population after dyn_GA calculation

We represent in Fig. 6(a) the predicted and the true spectra for the three types of neutron spectrum. We note a good agreement between the predicted and the real spectra with this method and for the different tested spectrum types. Fig. 6(b) shows the performance metric $Q_s$ as a function of all the ratios of the test dataset. We observe that $Q_s$ is varying in the interval [0.05, 0.07] and no specific correlation can be observed between the ratio (type of spectrum) and the quality of the prediction.




(a) (b)

**Fig. 6.** (a) True and predicted spectra (thermal(top), fast (middle) and mixed (bottom)); (b) $Q_s$ as a function of ratio.

## 6 Conclusion and prospects

In this paper, we proposed a comparison of two methods to generate a dataset for artificial neural networks training to reconstruct neutron spectra. The first one is based on probability distributions while the second is based on genetic algorithms (GA). We found that the most adequate solution is obtained using GA to build a large and balanced neutron spectra dataset. Starting from a simple SERPENT parametric model, dyn_GA generated an optimal fraction set of materials aimed at modifying the energy of the impinging neutrons after their emission to generate a balanced database of neutron spectra. Dyn_GA is based on a key parameter dynamically updated following the fraction set evolving. The built dataset, associated with SNAC2 reaction rates, is used to train and to test a CNN based on transpose convolutional layers aiming to unfold neutron spectra and compared to the other dataset. Thanks to the balance and the wideness of the dataset, the spectral quality ($Q_s$) of the tested samples is satisfactory (around 0.06) and the different types of neutron spectra are efficiently predicted. This promising building technique can be easily adapted to other nuclear contexts and domains (fusion, acceleration…) and is applicable to other detectors, which opens the way to a smart trend of finding new solutions in the nuclear fields. It should be noted that the GA fulfils the requirements of the SNAC system where we mainly need to determine the proportion between thermal, epithermal, and fast neutrons because it has an impact on the dose. To meet other needs where a finer knowledge of the spectrum structures (resonances) is required, the GA algorithm will have to be improved/sophisticated regarding the fitness function (combination of criteria) and the management of mutations. Another improvement perspective is to improve automaticity and adaptability of the GA algorithm.

Acknowledgment: the authors would like to thank Yann Richet for his valuable and constructive suggestions regarding the chosen methodology.

## References

[1] F.D Brooks, H Klein, Neutron spectrometry—historical review and present status, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 476, Issues 1–2, 2002, Pages 1-11,ISSN 0168-9002, https://doi.org/10.1016/S0168-9002(01)01378-X.].
[2] Bramblett, Richard L.; Ewing, Ronald I.; Bonner, T.W. (1960). "A new type of neutron spectrometer". *Nuclear Instruments and Methods*. **9** (1): 1–12.
[3] J.M. Gómez-Ros, R. Bedogni, C. Domingo, J.S. Eakins, N. Roberts, R.J. Tanner, Results of the EURADOS international comparison exercise on neutron spectra unfolding in Bonner spheres spectrometry, Radiation Measurements, Volume 153, 2022, 106755, ISSN 1350-4487, https://doi.org/10.1016/j.radmeas.2022.106755.).
[4] Martinez Blanca et al. 2020, A Neutron Spectrum Unfolding Code Based on Generalized Regression Artificial Neural Networks, EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, Volume 7, Issue 24.
[5] C. Cao, Q. Gan, J. Song, K. Yang, L. Hu, F. Wang, T. Zhou, An adaptive deviation-resistant neutron spectrum unfolding method based on transfer learning, Nuclear Engineering and Technology, **52**, 2452-2459 (2020).
[6] J. Wang, Z. Guo, X. Chen, Y. Zhou, Neutron spectrum unfolding based on generalized regression neural networks for neutron fluence and neutron ambient dose equivalent estimations, Applied Radiation and Isotopes, **154**, 108856 (2019).
[7] C. Cao, Q. Gan, J. Song, P. Long, B. Wua, Y. Wua, A two-step neutron spectrum unfolding method for fission reactors based on artificial neural network, Annals of Nuclear Energy, **139** (2020) 107219.

[8] Jie Wang, Yulin Zhou, Zhirong Guo, Haifeng Liu, Neutron spectrum unfolding using three artificial intelligence optimization methods, Applied Radiation and Isotopes, 147 (2019) 136–143.
[9] Goldberg, David E. and John H. Holland. “Genetic Algorithms and Machine Learning.” Machine Learning 3 (1988): 95-99.
[ref2GN] Holland, John H. “Genetic Algorithms.” Scientific American, vol. 267, no. 1, 1992, pp. 66–73.
[10] Bouhadida, Asmae Mazzi, Mariya Brovchenko, Thibaut Vinchon, Mokhtar Z. Alaya, Wilfried Monange, François Trompier, Neutron spectrum unfolding using two architectures of convolutional neural networks, Nuclear Engineering and Technology, 2023, ISSN 1738-5733, https://doi.org/10.1016/j.net.2023.03.025.
[11] Maha Bouhadida, Mariya Brovchenko, Thibaut Vinchon, Wilfried Monange, Francois Trompier. Neutron spectra reconstruction based on an artificial neural network trained with a large built dataset. ICRS 14/RPSD 2022 (14th International Conference on Radiation Shielding and 21st Topical Meeting of the Radiation Protection and Shielding Division), ANS, Sep 2022, Seattle, United States. ⟨hal-03900950⟩
[12] Leppänen, J., et al. (2015) "The Serpent Monte Carlo code: Status, development and applications in 2013." Ann. Nucl. Energy, 82 (2015) 142-150.
[13] Katoch, S., Chauhan, S.S. & Kumar, V. A review on genetic algorithm: past, present, and future. Multimed Tools Appl **80**, 8091–8126 (2021). https://doi.org/10.1007/s11042-020-10139-6
[14] A. Lambora, K. Gupta and K. Chopra, "Genetic Algorithm- A Literature Review," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 380-384, doi: 10.1109/COMITCon.2019.8862255.
[15] John McCall, Genetic algorithms for modelling and optimization, Journal of Computational and Applied Mathematics, Volume 184, Issue 1, 2005, Pages 205-222, ISSN 0377-0427, https://doi.org/10.1016/j.cam.2004.07.034.
[16] Tolstikhin, Ilya O., et al. "Mlp-mixer: An all-mlp architecture for vision." Advances in neural information processing systems 34 (2021): 24261-24272.
[17] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
[18] Scikit-learn documentation : https://scikit-learn.org/stable/whats_new/v1.2.html