Nuclear Engineering and Technology 56 (2024) 5175–5181

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

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

Original Article

A novel irradiation module for ANICCA fuel cycle code based on multi-task learning

Victor J. Casas-Molina [a,b,*], Nerea Aguilera-Gómez [a,c], Pablo Romojaro [a], Iván Merino-Rodríguez [d], Augusto Hernández-Solis [a]

[a] Belgian Nuclear Research Centre (SCK CEN), Boeretang 200, 2400, Mol, Belgium
[b] Ghent University–imec, Technologiepark-Zwijnaarde 126, 9052, Ghent, Belgium
[c] Universidad Politécnica de Madrid, José Gutiérrez Abascal 2, 28006, Madrid, Spain
[d] Catholic University of the Maule, Av. San Miguel 3605, Talca, Chile

ARTICLE INFO

Keywords:
Deep learning
ANICCA
Nuclear fuel cycle
MOX
Neural network
Spent fuel characterization

ABSTRACT

This paper introduces an alternative approach to irradiation modelling within the context of nuclear fuel cycle codes like ANICCA, the fuel cycle code developed by the Belgian Nuclear Research Centre (SCK CEN). The focus lies on upgrading the irradiation module by substituting the CRAM-based depletion calculations for a more flexible and innovative approach based on Multi-Task Learning (MTL). Utilizing data generated with SERPENT2 Monte Carlo simulations, two MTL neural networks are developed to surrogate irradiation processes of Uranium Oxide (UOX) and Mixed Oxide (MOX) fuels, respectively. MTL enables simultaneous learning of the evolution of the inventories for different observables – i.e., transuranium elements, fission products, minor actinides and fertile materials – offering improved predictive capabilities compared to non-MTL neural networks, as demonstrated in the cross-validation tests.

Hyperparameters of the models were found using Bayesian optimization, resulting in superior model performance compared to the old CRAM-based model. Verification against SERPENT2, used as a reference code for comparison, demonstrates the stability and accuracy of the MTL-based models, outperforming the original CRAM method for the majority of predicted isotopes.

1. Introduction

In response to energy crises and climate change, countries are reevaluating their energy policies, with nuclear fission emerging as a viable solution [1,2]. For this reason, it is necessary to conduct studies that analyze the impact of the different nuclear fuel cycle strategies, technologies and scenarios that can be adopted on country-level, for their subsequent development. However, managing the nuclear fuel cycle is a complex and costly endeavor where the application of nuclear fuel cycle codes can ease the problematic related to the scenario complexity. ANICCA [3] fuel cycle code was developed by the Belgian Nuclear Research Centre (SCK CEN) for that precise purpose. It is used to analyze various nuclear fuel cycle strategies for deployment, operation and decommissioning of nuclear reactor fleets.

The core of ANICCA is the irradiation module, which solves Bateman's transport equation to determine the final mass inventory of isotopes in spent nuclear fuel (SNF). Initially, this module used pre-computed one-group macroscopic cross-section libraries, generated by Monte Carlo (MC) codes such as ALEPH [4] and SERPENT2 [5], for irradiation calculations. These libraries allowed ANICCA to perform the burn-up process in a single step. The Chebyshev rational approximation method (CRAM) [6] was then implemented to solve Bateman's transport equations in a more recent version of the code, obtaining the isotopic vector at discharge.

However, this approach also had limitations: the cross sections in the libraries were averaged over time and burn-up, collapsed into one energy group, and the pre-built library solution was rigid. To address these issues and following similar work conducted on uranium based fuel [7], where a neural network used to predict the isotopic composition of the spent UNF fuel was implemented in the CYCLUS fuel cycle code [8], the irradiation process of both Uranium Oxide (UOX) and Mixed Oxide (MOX) fuels was initially surrogated via Deep Learning techniques following a so-called vanilla approach, i.e., using fully connected neural networks trained on a database generated by MC calculations [9].

* Corresponding author. Belgian Nuclear Research Centre (SCK CEN), Boeretang 200, 2400, Mol, Belgium.
E-mail address: victor.casas.molina@sckcen.be (V.J. Casas-Molina).

https://doi.org/10.1016/j.net.2024.07.024
Received 8 May 2024; Received in revised form 13 June 2024; Accepted 7 July 2024
Available online 8 July 2024

Nonetheless, the models were limited to 73 isotopes. Furthermore, the MOX predictions were constricted to a fixed Pu vector and finally, the importance of a given isotope for the fuel cycle was not considered during the training process of the models.

To overcome the aforementioned problem, this work proposes a new approach based on Multi-task learning (MTL) [10]. In MTL, the neural network architecture and the objective (loss) function are defined separately, based on different tasks to be simultaneously learned. These new models were trained and deployed in ANICCA to replace the former vanilla approach after a full verification study was conducted.

2. Materials and methods

2.1. Training datasets

Two different models based on neural networks were created to surrogate irradiation of UOX and MOX fuels, respectively. The data used for training the models was generated via irradiation simulations of the two different type of fuels performed with the SERPENT2 MC code [5]. The UOX dataset contains 63 531 samples of uranium oxide fuel ranging from 1.5 wt% to 6.0 wt% in steps of 0.025 wt% in Initial Enrichment (IE). The MOX dataset consists in 877 500 samples obtained by scaling 500 different Pu vector compositions (Pu + Am) by an Initial Plutonium Content (IPC) that spans from 4 wt% to 10 wt% in steps of 2 wt%. The Pu vector samples were derived from correlations for Reactor-Grade plutonium present in MOX fuel available in Ref. [11]. For the MOX case, the remaining metallic mass is depleted uranium (99.75 % of ^{238}U). Additionally, and for both datasets, ^{16}O is added subsequently to achieve a 2:1 atomic ratio of uranium or plutonium dioxide. Both datasets present a discharge burn-up (BU) ranging from 0 to 70 MWd/kg$_{HM}$ calculated in steps of 0.2 MWd/kg$_{HM}$, The datasets, the methods used for their generation, and the postprocessing techniques applied to the data can all be found and freely accessed in Ref. [12][dataset].

The utilized dataset contains the density estimates of 150 isotopes present in the SNF as a function of the initial isotopic content of the fuel and level of burn-up reached. Nonetheless, ^{16}O and ^{3}H were removed from the database prior to the training of both models since ^{3}H is a hard-to-predict isotope with a half-life of about 12 years, which means that its impact on the backend can be considered negligible [13]. ^{16}O was never considered in ANICCA, since calculations done by CRAM on fuel cycle codes are merely based on heavy metal content [14]. The remaining 148 isotopes were split into four different sets of observables: minor actinides (MA), transuranium elements (TRU), fertile material (FM) – i.e., uranium isotopes different from ^{235}U – and fission products (FP), which are the category received by the isotopes that cannot be included in one of the previous.

2.2. Neural networks

The UOX model surrogates the relationship between the IE of the uranium, the BU reached in the irradiation process and the final isotopic inventory, i.e., the different relative mass of the 148 isotopes at the end of the cycle.

On the other hand, the MOX model, in order to fully characterize the relationship between fresh fuel and discharge inventory, requires to account for the different combinations of plutonium isotopes in the initial vector of the fresh fuel. This includes the proportion of ^{238}Pu, ^{239}Pu, ^{240}Pu, ^{241}Pu, ^{242}Pu, and ^{241}Am, i.e., the plutonium vector, the BU and the IPC. Incorporating the content of ^{241}Am as an input allows the model to simulate the irradiation of MOX with different storage time, since the ^{241}Am builds up as the ^{241}Pu decays into it.

A detailed explanation of neural networks can be found in Ref. [15]. Neural networks perform regression by learning to map input features to continuous output values through a process called training. The network consists of an input layer, hidden layers, and an output layer. Each hidden layer neuron applies a weighted sum of its inputs followed by a non-linear activation function to model non-linear relationships in the data. During training, input data is fed through the network to produce predictions, which are compared to actual target values using a loss function.

The difference or distance (loss) is propagated back through the network via backpropagation, where gradients of the loss with respect to the network's weights and biases are computed. These gradients guide the adjustment of the weights and biases using optimization algorithms. By iteratively minimizing the loss function through this process, the neural network learns to produce accurate continuous outputs, effectively performing the regression task.

An essential concept in this training process is the epoch. An epoch refers to one complete pass through the entire training dataset. Training a neural network typically requires multiple epochs because a single pass through the data is usually insufficient for the network to learn the underlying patterns. During each epoch, the network weights are adjusted based on the gradients computed from the entire training dataset, enabling the network to gradually improve its performance. By using multiple epochs, the network can refine its weights and biases incrementally, leading to better convergence and improved accuracy in the regression task.

Another important concept is the batch size, which refers to the number of training samples processed before the model's internal parameters are updated. Instead of adjusting the weights and biases after each individual training example (which is computationally intensive and inefficient), the training dataset is divided into smaller groups called batches. Each batch is passed through the network to compute the loss and gradients, which are then used to update the network parameters. This process is repeated for all batches in an epoch. The batch size affects the training process by influencing the stability of the gradient estimates and the overall training time. Smaller batch sizes provide more precise gradient estimates but can lead to noisier updates, while larger batch sizes offer more stable updates but require more memory and can slow down convergence.

As an illustrative example, a schematic diagram of the UOX model can be seen in Fig. 1. The models have the neurons of each layer connected following a branching pattern, indicative of a Multi-task learning topology. In this topology, neurons in the initial common layers are densely connected, sharing information across all tasks (regression of the different sets of observables in the spent fuel). From these common layers, the network branches out into intermediate layers. Neurons in these intermediate layers are selectively connected, refining and processing information pertinent to subsets of tasks. Finally, the neurons in the intermediate layers diverge into task-specific layers, where they are densely connected within each task's domain, allowing for specialized output tailored to the individual tasks. This structured connection pattern - from densely connected common layers, through selectively

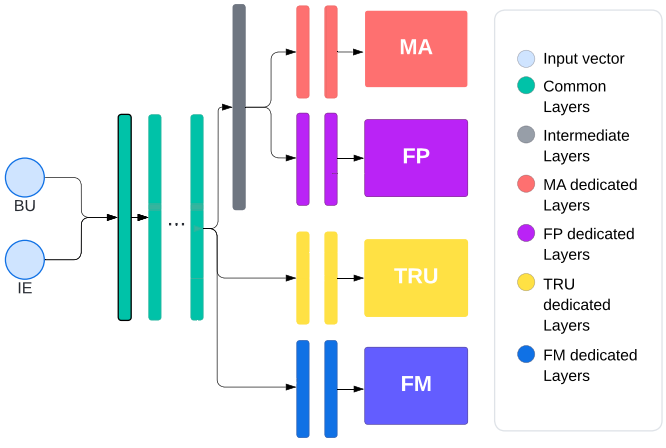

Fig. 1. Illustrative depiction of the UOX neural network.

connected intermediate layers, to densely connected task-specific layers – enables the model to balance shared learning and task-specific specialization effectively.

Additionally, two vanilla models for UOX and MOX were generated to serve as a baseline compare the performance of both approaches. These vanilla models are neural networks that are trained with the same dataset and with identical input and output dimensionality, the only difference is that all the layers are densely connected, without discriminating the output into groups.

All the models presented in this work were trained in Keras [16], a Python API for deep learning running on top of Tensorflow [17].

2.3. Model architecture and validation

In order to find the architecture of the model that performs the best, the determination of the so-called hyperparameters, such as the number of shared, intermediate and task-specific layers, number of neurons of each layer, kind of activation functions, etc., needs to be done before the training takes place. This process is normally known as hyperparameter tuning [15] and helps in deciding the best combination of topology and training settings for a given model in terms of performance. In order to conduct this tuning, the models are initialized with a different combination of neurons, layers and other options, such as the learning rate, which accounts for the size of the step taken during backpropagation of the gradient during training. Then, for each set of hyperparameters proposed, the model is trained with the dataset and the final score is recorded. Finally, the combination of hyperparameters chosen will be the one belonging to the model with the best performance explored during this hyperparameter tuning process. Prior to any hyperparameter tuning, it was determined to use Rectified Linear Unit (ReLU) activation functions for the hidden layers [18], since it is standard in regression problems. Hence, non-linearity of the underlying functions can be approximated. For the output layers, sigmoid functions were used to avoid the neural network returning negative values for the output vector. Additionally, the Huber loss (HBL) [19], defined as follows, was set:

$$HBL = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(y_i-\hat{y}_i)^2 & |y_i-\hat{y}_i| \le \delta \\ \frac{1}{n}\sum_{i=1}^{n}\delta|y_i-\hat{y}_i| - \frac{1}{2}\delta & |y_i-\hat{y}_i| > \delta \end{cases} \quad (1)$$

Where y_i is the observed value of the i-th of the n elements in the studied vector (the isotopic composition), and $\hat{y}_i$ the predicted one. HBL behaviour is balanced between the Mean Squared Error (MSE) and Mean Absolute Error (MAE) by the parameter delta (δ). The parameter δ acts as a threshold: when the distance between the real vector and the predicted one is smaller than δ, the loss function behaves as MSE, penalizing smaller errors more significantly. For larger errors, the loss function behaves more like MAE, penalizing the errors linearly, which reduces the impact of outliers on the training process. By setting the Huber loss, the neural network can handle both small and large errors effectively, providing a robust training process that balances sensitivity to outliers with precision for minor discrepancies. This makes the Huber loss particularly suitable for regression tasks where the data may contain noise or outliers. The loss function for the MTL was the equally weighted sum of the HBL values for each task, i.e., each group of observables, which means that the performance in the regression of each group is equally important.

Since training of neural networks is computationally expensive, the hyperparameter tuning was conducted by Bayesian optimization [20] using the Keras tuner [21]. In the Bayesian optimization framework, a surrogate model that relates the different hyperparameter combinations to a performance value (in this case, the validation loss of each fully trained model) is typically modelled using a Gaussian Process (GP). The GP provides a probabilistic model of the objective function, capturing the uncertainty about the function's shape and allowing efficient exploration of the hyperparameter space.

Then, the next candidates to be tested as better hyperparameters are chosen by optimizing an acquisition function, which balances exploration (trying new hyperparameter combinations) and exploitation (focusing on promising areas of the hyperparameter space). The acquisition function guides the selection process by predicting which hyperparameters are likely to improve model performance based on the surrogate model. This iterative approach continues until a predefined budget of trials is exhausted (in this work, 100 trials) or an optimal set of hyperparameters is identified, thereby optimizing the network's performance while minimizing the computational cost of hyperparameter search.

For the vanilla neural networks, the architecture was determined using the same hyperparameter tuning strategy to find the common layers and the number of neurons for each layer. The optimizer budget for the four models (i.e., UOX-Vanilla, UOX-MTL, MOX-Vanilla and MOX-MTL) was set in 100 epochs for a batch size of 32, using a training-test split ratio of 80/20 in the dataset. The training was conducted using the Adam optimizer [22]. Previous to the training, the weights of all the layers were initialized following the He normal distribution, since it was observed to work well when working with the utilized activation functions speeding up convergence [23].

Table 1 shows the bounds imposed for the hyperparameter tuning and the best combination found for each hyperparameter.

Table 1
Hyperparameter bounds and best configuration found from the models optimization.

MODEL	Hyperparameter	Bounds	Found parameters (UOX)	Found parameters (MOX)
MTL	common layers	[1,5]	1	1
	dedicated FP layers	[1,5]	2	2
	dedicated FM layers	[1,5]	2	2
	dedicated TRU layers	[1,5]	2	2
	dedicated MA layers	[1,5]	2	2
	neurons per common layer	[32, 512]	192	160
	neurons per FP layer	[32, 512]	288	480
	neurons per FM layer	[32, 512]	288	480
	neurons per TRU layer	[32, 512]	288	480
	neurons per MA layer	[32, 512]	288	480
	Huber delta (δ)	[0,1]	0.7986	0.0720
	learning rate	[1e-4,1e-2]	0.000776	0.001610
VANILLA	layers	[1,10]	4	4
	neurons per layer	[32, 512]	512	512
	Huber delta (δ)	[0,1]	0.8878	0.0699
	learning rate	[1e-4,1e-2]	0.001918	0.000238

2.4. Model selection

The four models were checked for both stability and accuracy by resorting to K-Fold cross-validation [24,25], which is a standard for model validation in machine learning. In K-Fold cross-validation, the final, best model found by the hyperparameter tuner is trained K times with a different division of the whole available dataset each time, by doing so, the model performance is tested in terms of the final results averaged over K: the standard deviation of the metrics informs about the model's average performance on unseen data, the average would be used to distinguish the model with the better accuracy. For this work, the four models were trained 1000 epochs 10 times (K = 10) using a batch size of 32. The metric chosen to compare the MTL models against the baseline (vanilla versions) is the MSE of the validation subset for each K, which was computed between the reference values and predicted values for the 148 isotopes without any weighing for each of the ten validation subsets.

2.5. ANICCA test case: the Belgian Phase Out

To evaluate the generated models, a testing scenario has been designed. It consists of an adaptation to the code's current capabilities of the current Belgian nuclear phase-out scenario, which is an updated version of the scenario described in Ref. [21]. The updated version of the phase-out accounts for the official life extension of the Doel 4 and Tihange 3 units until 2036 granted by the Belgian government. Beginning of final disposal was set at year 2100 as a reasonable approximation for the installation to start its operation. Wet and dry storage facilities were incorporated to represent the spent fuel facilities from the nuclear power plants. Uranium and plutonium pools were also added to the scenario as an intermediate stage where materials can be stored until its future use. Last facility worth-mentioning is NIRAS, which is the denomination used in this scenario for the temporary storage managed by the Belgian national agency for radioactive waste and enriched fissile materials management (ONDRAF/NIRAS). In this facility sub-products from other fuel cycle stages and installations, as well as everything that does not meet the requisites for being stored or processed in other interim storage facilities, are stored.

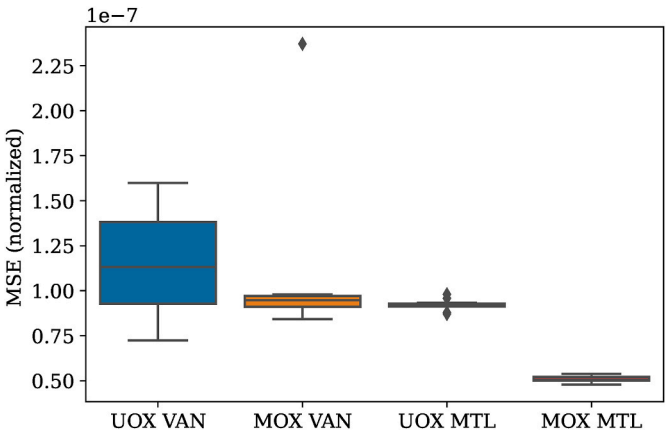

Fig. 3. Cross validation results of the models. VAN stands for VANILLA models and MTL for multi-task learning models. MSE is the mean squared error of the normalized values predicted by the models.

Fig. 2 depicts a full, comprehensive and simplified scheme of the cycle. Each box represents a facility, or in the case of the irradiation, a group of facilities, one for each reactor present in the scenario. Arrows, on the other hand, represent the mass flows between the different facilities. In this scenario, seven reactors are responsible for power generation. More technical information about each reactor can be found at [26].

The scenario begins with the production of natural uranium in the mine facility when required by the fabrication facility. The fabrication plant generates both the UOX and the MOX needed for the cycle that are transported to each reactor. Different enrichments for UOX fuel are required during the cycle and although it is not represented in the scheme for the sake of clarity, it is included in the output information. The spent fuel generated in each reactor is transported to the respective facility, with each spent fuel represented in a different color in the diagram. The orange arrows represent UOX fuel with a burn-up of 30 GWd/t_{HM}, the green arrows are for UOX with 45 GWd/t_{HM} burn-up and lastly, the reprocessed fuel (MOX), which is shown in blue.

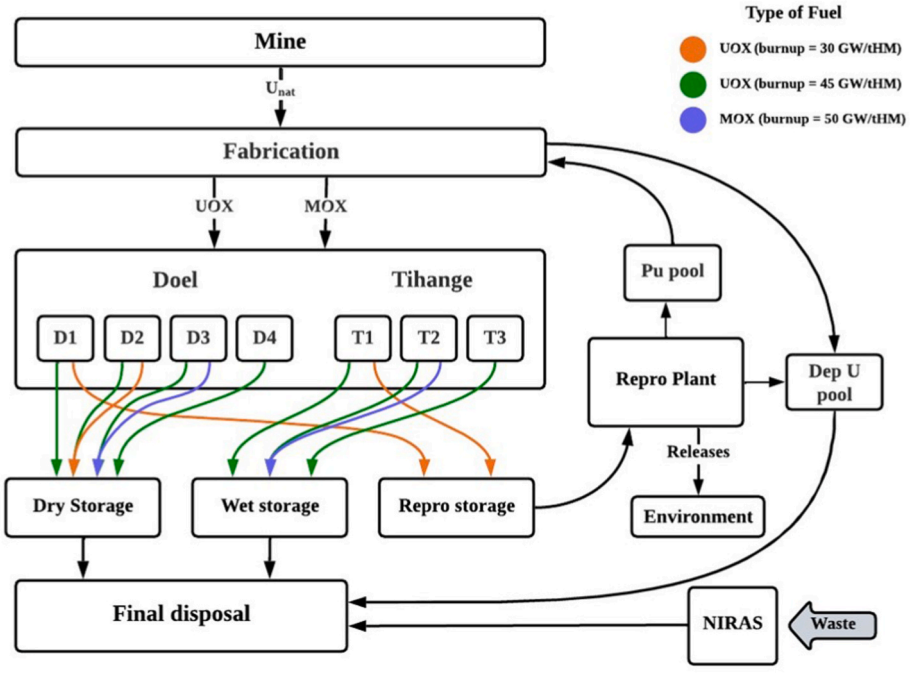

Fig. 2. Belgium Phase Out Scenario. For the sake of simplicity, neither reprocessed U or waste coming from that reprocessing are not included.

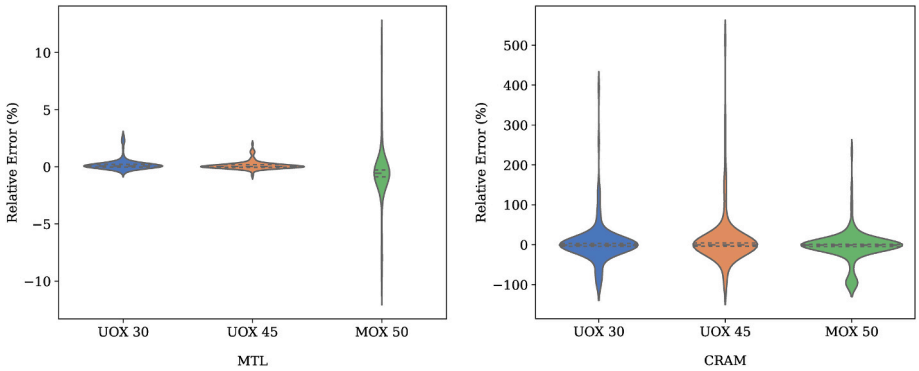

Fig. 4. Relative error for the 148 isotopes measured against the original SERPENT calculation for UOX burnt up to 30 and 45 GWd/tHM and MOX burnt at 50 GWd/tHM.

Table 2
Comparation of prediction relative error for CRAM and MTL versus SERPENT for some important isotopes. In bold the isotopes that are better predicted using the neural network approach than using CRAM.

	UOX 30		UOX 45		MOX 50	
	CRAM	MTL	CRAM	MTL	CRAM	MTL
^{90}Sr	−3.1 %	**−0.1 %**	−3.1 %	**−0.1 %**	−1.8 %	**−0.6 %**
^{137}Cs	−1.5 %	**0.0 %**	−1.2 %	**0.0 %**	−1.3 %	**−0.3 %**
^{148}Nd	1.1 %	**0.1 %**	1.0 %	**0.0 %**	0.0 %	−0.4 %
^{234}U	393.7 %	**−0.3 %**	512.6 %	**−0.6 %**	−56.0 %	**1.0 %**
^{235}U	4.5 %	**−0.4 %**	6.2 %	**−0.8 %**	−0.4 %	0.8 %
^{236}U	−0.2 %	**−0.2 %**	0.0 %	−0.2 %	−4.5 %	**−0.9 %**
^{238}U	0.0 %	**0.0 %**	−0.1 %	**0.0 %**	0.0 %	−0.1 %
^{237}Np	2.9 %	**−0.1 %**	3.1 %	**−0.1 %**	8.0 %	**−0.5 %**
^{238}Pu	0.7 %	**0.1 %**	1.7 %	**0.1 %**	−3.6 %	**3.4 %**
^{239}Pu	5.4 %	**−0.1 %**	6.2 %	**−0.1 %**	7.1 %	**3.1 %**
^{240}Pu	3.5 %	**0.1 %**	4.2 %	**0.1 %**	−0.6 %	**0.3 %**
^{241}Pu	4.3 %	**0.2 %**	5.4 %	**0.1 %**	3.0 %	**0.8 %**
^{242}Pu	−1.2 %	**0.4 %**	−0.3 %	**0.2 %**	−8.9 %	**−0.1 %**
^{244}Pu	113.3 %	**0.8 %**	77.4 %	**0.8 %**	18.4 %	**0.1 %**
^{241}Am	−1.6 %	**0.1 %**	−0.6 %	**0.0 %**	0.8 %	10.1 %
^{243}Am	0.6 %	0.8 %	0.8 %	**0.6 %**	5.2 %	**−0.3 %**
^{242}Cm	3.2 %	**0.3 %**	6.1 %	**0.1 %**	2.3 %	10.7 %
^{244}Cm	2.8 %	**0.7 %**	3.4 %	**0.6 %**	10.2 %	**−0.5 %**
^{245}Cm	8.9 %	**1.0 %**	10.3 %	**0.7 %**	16.4 %	**−0.5 %**
^{246}Cm	2.2 %	**1.6 %**	3.4 %	**1.1 %**	10.5 %	**−0.5 %**
^{248}Cm	−0.3 %	2.6 %	−3.0 %	**1.2 %**	−0.4 %	1.5 %
^{249}Bk	77.5 %	**2.7 %**	94.2 %	**2.0 %**	50.1 %	**2.8 %**
^{249}Cf	129.0 %	**2.4 %**	178.6 %	**1.4 %**	110.0 %	**4.2 %**
^{250}Cf	−100.0 %	**2.1 %**	−100.0 %	**1.5 %**	−100.0 %	**4.5 %**
^{251}Cf	−100.0 %	**2.1 %**	−100.0 %	**1.3 %**	−100.0 %	**4.9 %**

The first stage of the back end of the cycle, the temporary storage of the fuel, is slightly different from reality. The reprocessing storage is an intermediate facility that is not real but used to store fuel that will be reprocessed later. Dry storage is used at the Doel nuclear power plant. All the spent fuel that the plant generates during its operation arrives there, with the exception of spent fuel with a burn-up of 30 GWd/t_{HM}, which is sent to the intermediate storage for further reprocessing. Pool or wet storage, on the other hand, is where all the spent fuel from the Tihange nuclear power plant is located. From dry and wet storage, only one last step is currently envisaged in the Belgian stage, final storage.

3. Results and discussion

3.1. Validation of the models

The results of the cross validation test of the four models can be found in Fig. 3, where each box plot collectively represents the 10 different validation results of each model.

As can be observed, the vanilla models behave worse in terms of both stability and accuracy. Big spreads in the different evaluations can be seen for the UOX vanilla model, additionally a great outlier can be observed for one of the folds tested in the MOX vanilla model.

On the other hand, both MOX and UOX MTL-based models have less spread for all the different tested folds and a smaller average value that their respective vanilla models, indicating a best general performance of the MTL approach with respect to the simpler densely connected vanilla approach when comparing to the original training data.

Once the stability of the models was checked, the same fuel cases used in the Phase Out Scenario, described in subsection 2.4, were simulated using the MTL models and the original CRAM method included in the original ANICCA irradiation module. The relative error of the 148 predicted isotopes was calculated against the original SERPENT2 calculations and is represented in Fig. 4 for the new MTL models (left) and the original CRAM (right).

As can be appreciated and for all the fuel types, the relative error is about two orders of magnitude smaller for the MTL approach with respect to the CRAM-based method. For the MTL approach, the maximum deviations are observed for the MOX fuel.

Nonetheless, as can be observed in Table 2, the majority of isotopes of interest for both spent fuel characterization and spent fuel analysis are predicted with less than 10 % of relative error.

As can be observed, when checking against SERPENT2, CRAM predicts systematically with greater deviation the majority of the isotopes, with rare exceptions.

3.2. Use case: the Belgian Phase Out

The Belgian Phase Out Scenario described in subsection 2.4. was simulated in ANICCA with two different implementations for the irradiation module: the traditional CRAM and the neural network based (MTL). Given that the results presented in Section 3.1 indicated that the MTL approach outperforms the traditional vanilla approach, it was decided to not evaluate the latter for this specific use case.

In Fig. 5, the evolution of the total uranium, FP, TRU and MA in the spent fuel inventory for the Belgian Phase Out can be appreciated. Additionally, some relevant isotopes of each group are presented with dashed lines.

As can be observed, the behavior of the two implementations matches, showing a good coupling of the MTL models with the rest of the code in ANICCA. The difference of the values between MTL and CRAM can be attributed to the differences of individual isotopes with their reference SERPENT2 values. In total, over the period observed in the simulation for all Belgium, the variation of the observables predicted by the MTL over the CRAM are about 4.6 %, 1.4 % and 4.0 % for TRU, FP and MA respectively, whereas there is no difference for the total uranium.

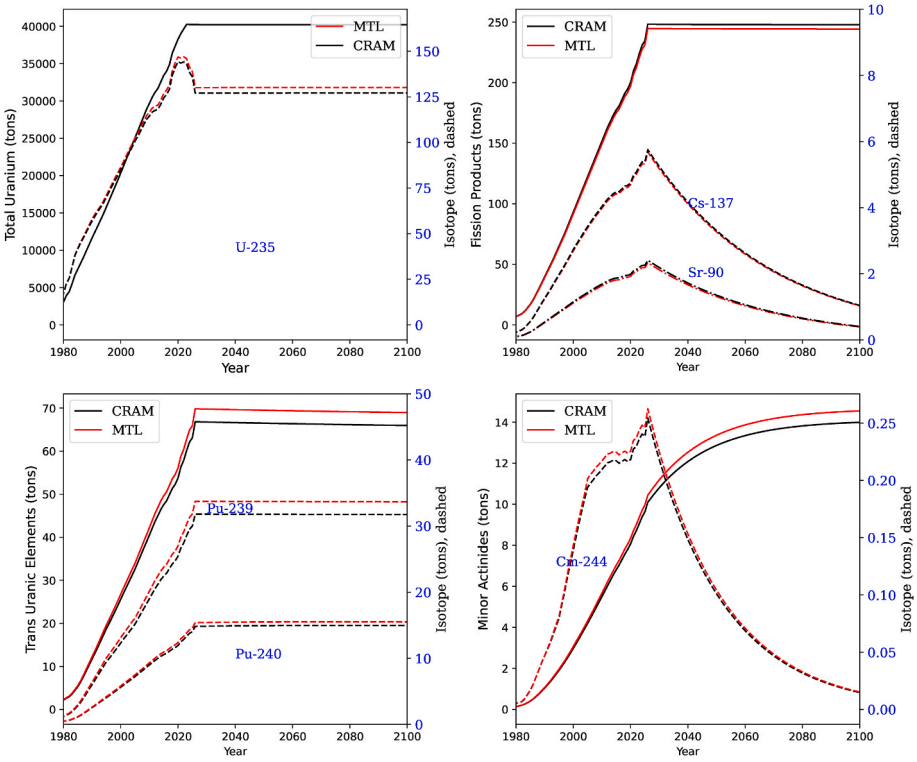

Fig. 5. Evolution of the inventory of spent fuel simulated for the Belgian Phase Out scenario.

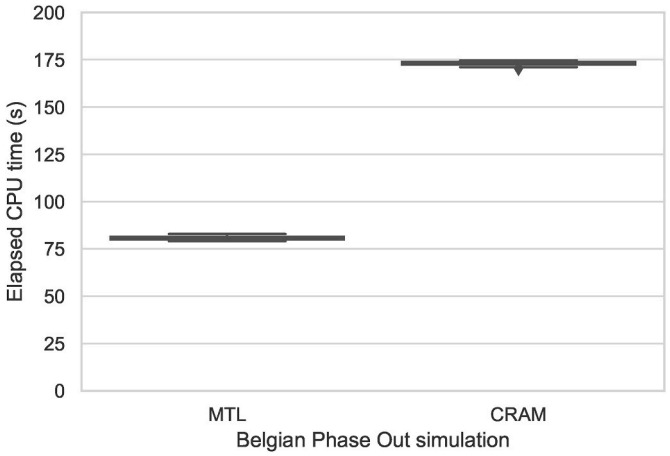

Fig. 6. Elapsed CPU time in seconds for the Belgian Phase Out scenario.

Additionally, and in order to assess the elapsed calculation time for both methods, the introduced use case has been run 40 times to produce a comparative study of the CRAM and MTL runtimes. The results of this evaluation are shown in Fig. 6.

As can be observed, the speed of the MTL implementation is up to 2.3 times higher than the CRAM approach when resolving complex scenarios such as the present use case.

4. Conclusion

In this work, a method to surrogate the simulated irradiation processes for all the possible UOX and MOX fuel compositions in PWRs by using neural networks trained using a Multi-task learning approach has been presented. Moreover, it has been proved that the MTL models outperform their vanilla counterparts in such task, delivering systematically better results. Additionally, the implementation allows to simulate any composition of any given MOX or UOX fuel within the application range and without depending from another codes anymore, since no burn-up libraries are required any longer. At the same time, the speed of the newer implementation cuts down the runtime by half.

Finally, it can be concluded that the MTL models were satisfactory deployed in the ANICCA fuel cycle code and that they outperform the traditional CRAM-based method when tested against the SERPENT2 Monte Carlo code, used as a reference.

CRediT authorship contribution statement

Victor J. Casas-Molina: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Nerea Aguilera-Gómez:** Writing – original draft, Software. **Pablo Romojaro:** Writing – review & editing, Validation, Supervision, Resources, Project administration. **Iván Merino-Rodríguez:** Writing – review & editing, Methodology. **Augusto Hernández-Solis:** Writing – review & editing, Supervision, Resources, Conceptualization.

Declaration of competing interest

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

References

[1] T. Tajima, A. Necas, T. Massard, S. Gales, East meets West again in order to tackle the global energy crises, Usp. Fiz. Nauk 192 (11) (2022) 1280–1292, and others.
[2] G. Halkos, A. Zisiadou, Energy crisis risk mitigation through nuclear power and RES as alternative solutions towards self-sufficiency, J. Risk Financ. Manag. 16 (1) (Jan. 2023) 1, https://doi.org/10.3390/jrfm16010045.
[3] I.M. Rodríguez, A. Hernández-Solís, N. Messaoudi, G. Van den Eynde, The nuclear fuel cycle code ANICCA: verification and a case study for the phase out of Belgian nuclear power with minor actinide transmutation, Nucl. Eng. Technol. 52 (10) (Oct. 2020) 2274–2284, https://doi.org/10.1016/J.NET.2020.04.004.

[4] A. Stankovskiy, G. Van den Eynde, Advanced method for calculations of core burn-up, activation of structural materials, and spallation products accumulation in accelerator-driven systems, Sci. Technol. Nucl. Install. 2012 (Apr. 2012) e545103, https://doi.org/10.1155/2012/545103.
[5] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho, The Serpent Monte Carlo code: status, development and applications in 2013, Ann. Nucl. Energy 82 (Aug. 2015) 142–150, https://doi.org/10.1016/J.ANUCENE.2014.08.024.
[6] M. Pusa, Rational approximations to the matrix exponential in burnup calculations, Nucl. Sci. Eng. 169 (Oct. 2011) 155–167, https://doi.org/10.13182/NSE10-81.
[7] J.W. Bae, A. Rykhlevskii, G. Chee, K.D. Huff, Deep learning approach to nuclear fuel transmutation in a fuel cycle simulator, Ann. Nucl. Energy 139 (May 2020) 107230, https://doi.org/10.1016/j.anucene.2019.107230.
[8] K.D. Huff, et al., Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework, Adv. Eng. Software 94 (Apr. 2016) 46–59, https://doi.org/10.1016/j.advengsoft.2016.01.014.
[9] V.J.C. Molina, A.H. Solís, I.M. Rodríguez, P.R. Otero, Deep learning models as an approach to nuclear fuel irradiation processes in pressurized water reactors, in: 2022 41st International Conference of the Chilean Computer Science Society (SCCC), Nov. 2022, pp. 1–8, https://doi.org/10.1109/SCCC57464.2022.10000327.
[10] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv (2017), https://doi.org/10.48550/arXiv.1706.05098. Jun. 15.
[11] I.C. Gauld, MOX Cross-Section Libraries for ORIGEN-ARP, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States), Jul. 2003, https://doi.org/10.2172/885544. ORNL/TM-2003/2.
[12] V.J. Casas-Molina, A. Hernandez-Solis, P. Romojaro, I. Merino-Rodríguez, N. Aguilera-Gómez, Dataset of observables for UOX and MOX spent fuel extracted from Serpent2 fuel depletion calculations for PWRs, Data Brief 49 (Aug. 2023) 109412, https://doi.org/10.1016/j.dib.2023.109412.
[13] RED-IMPACT Impact of partitioning, transmutation and waste reduction technologies on the final nuclear waste disposal. Synthesis report, in: Werner von Lensa, Rahim Nabbi, Matthias Rossbach (Eds.), Schriften des Forschungszentrums Jülich Reihe Energie & Umwelt, vol. 15, Germany, 2008. ISBN 978-3-89336-538-8.
[14] NEA (2012), Benchmark Study on Nuclear Fuel Cycle Transition Scenarios Analysis Codes, OECD Publishing, Paris. Accessed: April. 14, 2023. [Online]. Available: https://www.oecd-nea.org/jcms/pl_19182/benchmark-study-on-nuclear-fuel-cycle-transition-scenarios-analysis-codes?details=true.
[15] D.A. Roberts, S. Yaida, B. Hanin, The Principles of Deep Learning Theory, 2022, https://doi.org/10.1017/9781009023405.
[16] F. Chollet and others, "Keras." [Online]. Available: https://github.com/fchollet/keras.
[17] Martín Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous systems [Online]. Available: https://www.tensorflow.org/, 2015.
[18] A.F. Agarap, Deep learning using rectified linear units (ReLU), arXiv, Feb. 07 (2019), https://doi.org/10.48550/arXiv.1803.08375.
[19] K. Gokcesu, H. Gokcesu, Generalized huber loss for robust learning and its efficient minimization for a robust statistics, arXiv, Aug. 28 (2021), https://doi.org/10.48550/arXiv.2108.12627.
[20] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, S.-H. Deng, Hyperparameter optimization for machine learning models based on bayesian optimization, J. Electron. Sci. Technol. 17 (1) (Mar. 2019) 26–40, https://doi.org/10.11989/JEST.1674-862X.80904120.
[21] T. O'Malley, et al., KerasTuner [Online]. Available: https://github.com/keras-team/keras-tuner, 2019.
[22] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv, Jan. 29 (2017), https://doi.org/10.48550/arXiv.1412.6980.
[23] L. Datta, A survey on activation functions and their relation with xavier and He normal initialization, arXiv (2020), https://doi.org/10.48550/arXiv.2004.06632. Mar. 18.
[24] B.G. Marcot, A.M. Hanea, What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? Comput. Stat. 36 (3) (Sep. 2021) 2009–2031, https://doi.org/10.1007/s00180-020-00999-9.
[25] D.M. Allen, The relationship between variable selection and data agumentation and a method for prediction, Technometrics 16 (1) (Feb. 1974) 125–127, https://doi.org/10.1080/00401706.1974.10489157.
[26] International Atomic Energy Agency (IAEA), Operating experience with nuclear power stations in member states 2022, IAEA Annual reports 53 (2022) 42–68.