Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

Deep learning approach to nuclear fuel transmutation in a fuel cycle simulator

Jin Whan Bae [*], Andrei Rykhlevskii, Gwendolyn Chee, Kathryn D. Huff

Dept. of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

ARTICLE INFO

Article history:
Received 3 August 2019
Received in revised form 18 November 2019
Accepted 19 November 2019
Available online 13 December 2019

Keywords:
Nuclear fuel cycle
Machine learning
Artificial neural network
Simulation
Spent nuclear fuel

ABSTRACT

We trained a neural network model to predict Pressurized Water Reactor (PWR) Used Nuclear Fuel (UNF) composition given initial enrichment and burnup. This quick, flexible, medium-fidelity method to estimate depleted PWR fuel assembly compositions is used to model scenarios in which the PWR fuel burnup and enrichment vary over time. The Used Nuclear Fuel Storage, Transportation & Disposal Analysis Resource and Data System (UNF-ST&DARDS) Unified Database (UDB) provided a ground truth on which the model trained. We validated the model by comparing the U.S. UNF inventory profile predicted by the model with the UDB UNF inventory profile. The neural network yields less than 1% error for UNF inventory decay heat and activity and less than 2% error for major isotopic inventory. The neural network model takes 0.27 s for 100 predictions, compared to 118 s for 100 Oak Ridge Isotope GENeration (ORIGEN) calculations.

We also implemented this model into CYCLUS, an agent-based Nuclear Fuel Cycle (NFC) simulator, to perform rapid, medium-fidelity PWR depletion calculations. This model also allows discharge of batches with assemblies of varying burnup.

Since the original private data cannot be retrieved from the model, this trained model can provide open-source depletion capabilities to NFC simulators. We show that training an artificial neural network with a dataset from a complex fuel depletion model can provide rapid, medium-fidelity depletion capabilities to large-scale fuel cycle simulations.

Published by Elsevier Ltd.

1. Introduction

1.1. Background and motivation

The NFC is a complex system of facilities and material mass flows that combine to provide nuclear energy, usually in the form of electricity (Yacout et al., 2005). NFC simulators are system analysis tools used to investigate issues related to the dynamics of a nuclear fuel cycle in both high and low-resolution. An example of a high-resolution element is the spent fuel isotopic composition from a single fuel bundle, and an example of a low-resolution element is the total fuel utilization in the system. The intention behind the use of NFC simulators is to develop a better understanding of the dependence between various components in the system and the effects of changes on the system. Their goal is to assist in evaluating and improving potential strategies for nuclear power development in terms of improving waste management, economic competitiveness, etc. (Yacout et al., 2005).

One of the major functionalities of an NFC simulator is its ability to transmute nuclear fuel in a reactor based on reactor conditions such as burnup, enrichment, etc. The transmutation results impact the accuracy of the UNF composition, and thus the capability of using the NFC to analyze the impact of an NFC on variables such as waste profile.

Current fuel cycle simulators include CYCLUS (Huff et al., 2016), DYMOND (Yacout et al., 2005), VISION (Jacobson et al., 2010), ORION (Gregg and Grove, 2012), COSI (Coquelet-Pascal et al., 2015), and CLASS (Mouginot et al., 2012), NFC simulators obtain transmutation results using various methods. Four types of methods to obtain transmutation results exist: recipe-based, library-based, spectral, and dynamic coupling methods. In recipe-based methods, direct neutronics calculations are not performed within the model, they are done externally (Yacout et al., 2006). The user inputs resulting fresh and spent fuel compositions for specific parameters directly into the NFC model (Sunny et al., 2015). In the library-based method, the NFC simulator dynamically calculates depleted fuel recipes by interpolating reactor data libraries

[*] Corresponding author at: Oak Ridge National Laboratory, 1 Bethel Vlley Road, Oak Ridge, TN, 37831, USA.

E-mail address: nuclearbae@gmail.com (J.W. Bae).

https://doi.org/10.1016/j.anucene.2019.107230
0306-4549/Published by Elsevier Ltd.

generated by reactor physics burnup calculation code such as SERPENT (Leppanen et al., 2015), Oak Ridge Isotope GENeration (ORIGEN) (Croff, 1980), etc. In spectral methods, a reactor model that incorporates spectral changes as a function of burnup is used (Scopatz et al., 2011). In dynamic coupling methods, a Fuel Cycle Simulator (FCS) is coupled with a reactor physics depletion code and a depletion calculation is conducted during an NFC simulation to obtain time-dependent transmutation results.

Table 1 shows a simple breakdown of the transmutation methods available for each NFC code.

CYCLUS has three methods to obtain spent fuel compositions: the CYCAMORE recipe reactor (Huff et al., 2014), CyBORG (Skutnik et al., 2016), and Bright-lite (Flanagan, 2014). The CYCAMORE recipe reactor accepts a fresh and spent fuel recipe that is defined by the user. CyBORG uses ORIGEN to dynamically calculate depleted fuel recipes by interpolating ORIGEN reactor data libraries to problem-specific conditions including initial enrichment, burnup, and user-specified interpolation parameters (Skutnik et al., 2016). Bright-lite uses an interpolated transmutation matrix approach with pre-configured libraries available for multiple commercial reactor types (Flanagan, 2014). This is a spectral type transmutation method. ORION has a recipe-based and library-based method to obtain transmutation results. In ORION's library-based method, burnup-dependent cross-section libraries for multiple reactor types with multiple initial fuel enrichment are generated before ORION analysis (Sunny et al., 2015). The libraries are then used to generate spent fuel recipes based on reactor conditions such as burnup, enrichment, etc. DYMOND has a recipe-based method in which recipes for both the input and output fuel compositions are externally calculated. Isotopes are tracked in lumped categories: fission products, minor actinides, uranium and plutonium (Feng et al., 2016). VISION has a recipe-based method with a limited set of individually tracked isotopes (Yacout et al., 2006).

For NFC simulators, striking a balance between fidelity and computational cost is a key issue. The advantage of using high fidelity models (dynamic coupling methods) is their inherent flexibility to readily accommodate varying fuel compositions when modeling complex scenarios (Sunny et al., 2015). However, using high fidelity models for century-long simulations can result in impractical computational times. The advantage of using low fidelity models (recipe-based methods) is the low computational cost. They are acceptable methods for modeling fuel cycles with fixed input composition or fuel cycles at equilibrium (Sunny et al., 2015). However, for fuel cycles not at equilibrium, they result in less accurate results.

Therefore, to find a middle ground of accurate depletion data while maintaining a practical computational cost, this paper introduces a trained dense neural network model that is able to predict PWR UNF composition based on initial enrichment and burnup.

1.2. Previous work

In 2015, Leniau et al. (2015) used neural networks for modeling Mixed Oxide Fuel (MOX) in the NFC simulator CLASS. They successfully demonstrated an application of neural networks to predict plutonium fraction in a fresh MOX fuel required to reach a specific burnup, as well as to predict mean cross section for depletion calculations. This work differs from the work by Leniau et al., since it predicts Uranium Oxide Fuel (UOX) fuel depletion, and the neural network predicts the depleted compositions directly, which eliminates the need to solve the Bateman equations. Also, we further extend the concept pioneered by Leniau et al. by using the trained neural network to predict a time-aggregated UNF inventory, and comparing the inventory with a pre-existing database. Lastly, this work implements the neural network to a reactor model to simulate dynamic (burnup, enrichment) reactor behavior.

Table 1
Methods for transmutation in reactor modules for each NFC simulation code.

NFC code	Transmutation Methods Available
Cyclus (Huff et al., 2016)	Recipe-based, library-based, spectral, and dynamic-coupling
ORION (Gregg and Grove, 2012)	Recipe-based and library-based
DYMOND (Yacout et al., 2005)	Recipe-based
VISION (Jacobson et al., 2010)	Recipe-based
COSI (Coquelet-Pascal et al., 2015)	Recipe-based, library-based, and dynamic coupling
CLASS (Mouginot et al., 2012)	Library-based

1.3. CYCLUS

CYCLUS is an agent-based nuclear fuel cycle simulation framework (Huff et al., 2016), meaning that each reactor and fuel cycle facility is modeled as a discrete and independent player in the simulation. A CYCLUS agent archetype defines the logic that governs the behavior of an agent. CYCLUS archetypes are implemented in either C++ or Python. In a simulation, the user defines the archetype's parameters. The archetypes with user-defined parameters are then deployed as agent prototypes. Encapsulating the `Facility` agents are the `Institution` and `Region`. A `Region` agent holds a set of `Institution`s. An `Institution` agent can deploy or decommission `Facility` agents.

At each timestep, agents make requests for materials or bid to supply them and exchange with one another. A market-like mechanism called the Dynamic Resource Exchange (Gidden and Wilson, 2016) governs the exchanges. For output analysis, each material resource has a quantity, composition, name, and a unique identifier.

Cyclus has multiple advantages over other available NFC simulation codes including open-source distribution, modularity, and extensibility. Its agent-based modeling approach is ideal for modeling coupled, physics-dependent supply chain problems common in NFCs. The framework allows for dynamic loading of external libraries, so that users can plug-and-play various physics models for NFC facility processes (shown in Fig. 1).

1.3.1. Modularity and extensibility

In most modern NFCs simulators, the facilities and their behaviors are confined in the software. Also, typical NFC simulators model fuel cycles (e.g. once-through, continuous reprocessing) with immutable connections between facilities. On the other hand, facilities in CYCLUS are not limited in their connections, due to its modular framework. This enables CYCLUS to simulate any system involving multiple connected facilities with physics-based calculations.

Due to this modularity in the CYCLUS framework, the developed model in this work is implemented independently without having to modify the CYCLUS source code. The new facility archetype is written in Python and implemented through the CYCLUS API.

2. Method

This work follows four steps:

1. Data curation
2. Model training (with hyperparameter optimization)
3. Model validation
4. Model implementation in CYCLUS

First, we curated the assembly information data for ease of use in training the model (Section 2.1.1). Second, we trained the dense

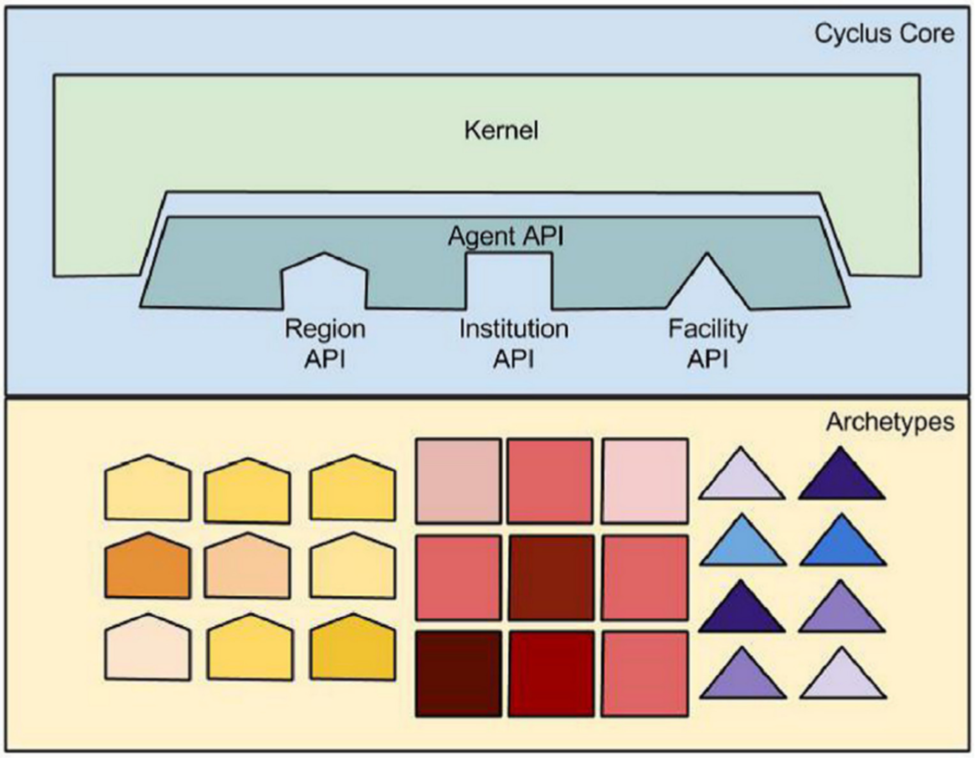

Fig. 1. The CYCLUS core provides APIs that the archetypes can be loaded into the simulation modularly (Huff et al., 2016).

neural network using Keras (Chollet et al., 2015) and Scikit-learn (Pedregosa et al., 2011). This workflow incorporated an outer loop to search for the optimized set of neural network *hyperparameters*, such as the number of hidden layers and nodes per layer. Third, we used the model to predict the U.S. UNF inventory as specified in the UDB and compare UNF inventory metrics such as fissile content and decay heat. Lastly, we implemented the trained model in CYCLUS by developing a reactor facility archetype that transmutes fuel using the model.

The files used to generate and test the dense network model are all on Github (Bae et al., 2019). The raw data is not available to the public.

2.1. Training set

In order to train an artificial neural network model, a significant database of depletion data is needed that spans the potential burnup and enrichment range in the reactor types involved in the fuel cycle simulation.

Data from the UDB was used to train the model based on burnup and enrichment.

We simply used all the PWR datasets in the UDB, and the input values are only burnup and initial enrichment. Ideally, the data should be generated with the same reactor parameters other than burnup and enrichment, such as lattice geometry. However, the database is generated with varying assembly geometries.

2.1.1. Unified database

The UDB [1] is part of a larger engineering analysis tool, the Used Nuclear Fuel Storage, Transportation & Disposal Analysis Resource and Data System (UNF-ST&DARDS), developed by Oak Ridge National Laboratory (ORNL) (Peterson et al., 2013). The database provides a comprehensive, controlled source of spent nuclear fuel (SNF) information, including dry cask attributes, assembly data, economic attributes, transportation infrastructure attributes, potential future facility attributes, and federal government radioactive waste attributes. The assembly-specific attributes include initial enrichment, burnup, metric tons of heavy metal (MTHM), assembly type, and discharge date (Peterson and Scaglione, 2015). To generate this database, the authors used irradiation and decay calculations using SCALE (Bowman, 2011). The calculations were performed on each spent fuel assembly based on the previously mentioned parameters in the collected data to obtain mass, heat, and activity for each assembly (Peterson et al., 2017). All the assemblies were modeled with conservative depletion parameters which result in the hardening of the neutron energy spectrum and an increased SNF residual reactivity (Peterson et al., 2017).

Also, the irradiation history of the fuel is unspecified in the database, which can be a source of deviation for short-lived isotopes. With the unknown parameters (unknown irradiation history, varying assembly models) and assumptions (conservative composition to increase fuel reactivity), the database is far from ideal to use as a training dataset for a depletion calculation model. However, we chose this data set because it allows testing of model performance through comparison of UNF inventory between a high-fidelity model and a model prediction for varying burnup, enrichment, and discharge time.

2.2. Data curation

We curated the raw UDB datasets to generate a cleaner training set. First, we only used the PWR assemblies since Boiling Water Reactor (BWR) UNF assembly calculation results can vary significantly with void fraction. We also filtered out the 'very low' enrichment ($\leqslant 1.5$) and burnup ($\leqslant$10,000 MWd/MT) assemblies to represent a more modern PWR UNF assembly range. Fig. 2 shows

[1] We received the database through personal communication with Dr. Kaushik Banerjee (ORNL).

the burnup and enrichment distribution of the assemblies in the UDB.

Notably, since the SCALE calculations in the UDB only track 60 isotopes, 3.5 weight percent of the UNF is unaccounted for, on average. We aggregate the isotopes not accounted for as a separate category. Lastly, we processed the database so that the isotopic compositions are represented as weight % normalized by initial uranium mass. For every isotope i:

$$x_i = \frac{m_i}{M_{initU}} \tag{1}$$

where:

x_i = Percent weight of isotope in depleted assembly
m_i = Mass of isotope in depleted assembly in UDB
M_{initU} = Mass of initial uranium in assembly

2.3. Predictive models for fuel depletion

UNF depleted composition prediction is complex due to the varying relationship with the fuel parameters. In Figs. 3–5, we

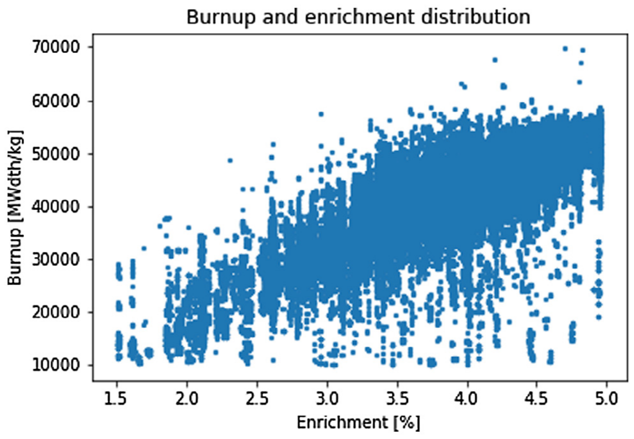

Fig. 2. Burnup and enrichment distribution of training datasets curated from the UDB.

observe the relationship between burnup, enrichment, and isotopic composition.

We observed that, if the isotopic population is mainly determined by the fission of initial uranium, a linear regression algorithm can be used to predict the isotopic composition from burnup (^{137}Cs shown in Fig. 3). However, isotopes like ^{239}Pu (Fig. 4) have multiple, conflicting creation and destruction terms, making it harder to predict using a linear regression algorithm. Also, the ^{235}U (Fig. 5) composition depends on both burnup and enrichment, which can make it hard to predict using a simple linear model.

Due to these complexities, we decided to train an artificial dense neural network for our predictive model. We chose Keras (Chollet et al., 2015) to create and validate the model, as well as scikit-learn (Pedregosa et al., 2011) and pandas (McKinney, 2010) for data processing and management.

2.4. Training and selecting models

The inputs (features) of the model are burnup (MWd/MT) and initial enrichment (wt% ^{235}U). The outputs (targets) of the model are the composition (weight %) of the 60 isotopes in the depleted assembly.

With the curated dataset, we performed an outer loop search to find the best-performing neural network hyperparameter (Table 2). First, we set aside 20 percent of the data for final model testing purposes. We used threefold cross validation (Stone, 1974) on the remaining dataset to measure the average prediction error value. We normalized the data using the `sklearn` MinMaxScaler so that the range of input and output data was (0,1).

We selected the model with the smallest average error value and exported the model as a Python pickle file along with the dataset normalization objects and the list of isotopes. By exporting the trained model as a self-contained file, the model can be used in any Python application.

2.5. Model testing

We tested the accuracy of the model by comparing its UNF composition prediction in three different cases. First, we compared the isotope-by-isotope prediction error of the model for an assembly

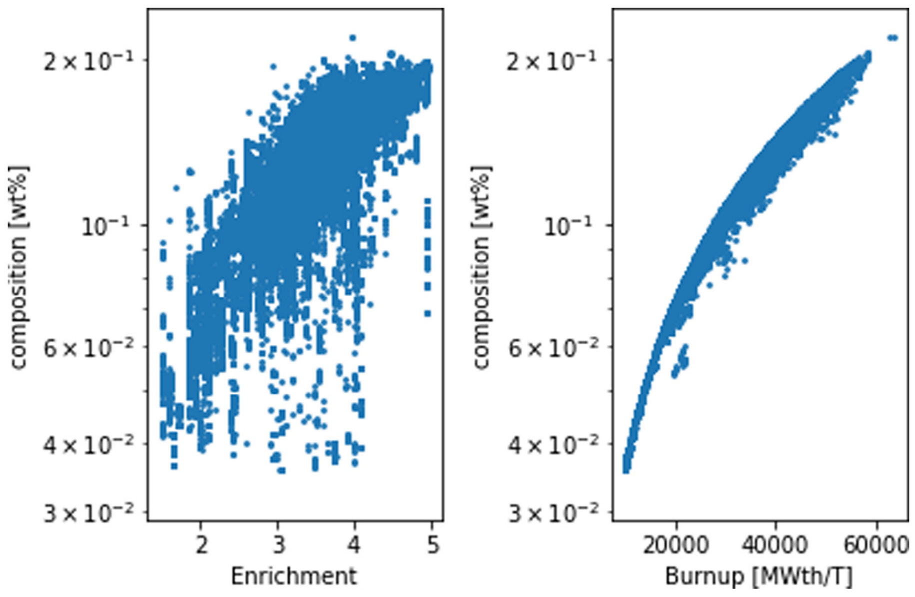

Fig. 3. ^{137}Cs composition in a UNF assembly varies linearly with assembly burnup.

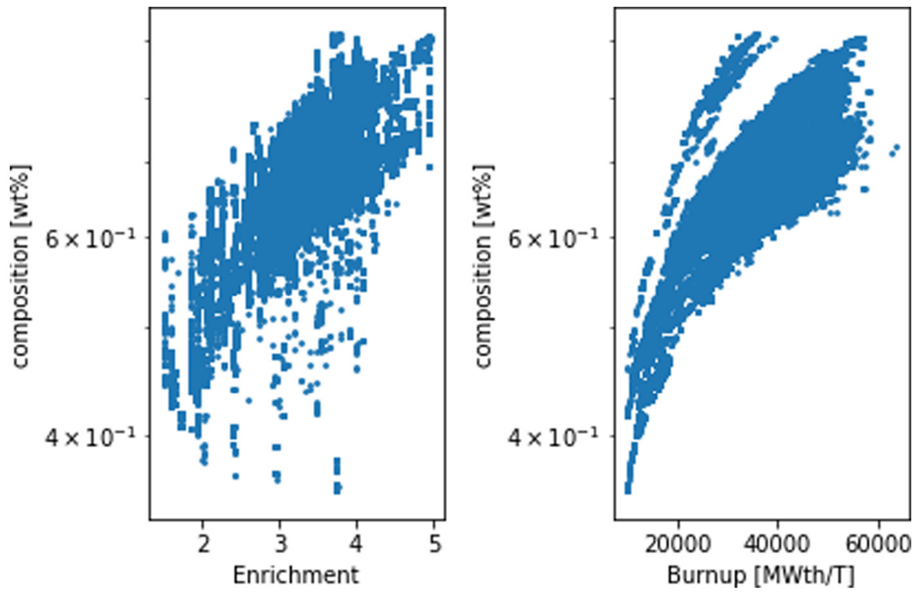

Fig. 4. ^{239}Pu composition in a UNF assembly is not linearly related to burnup, since it is affected by multiple, conflicting creation and destruction terms.

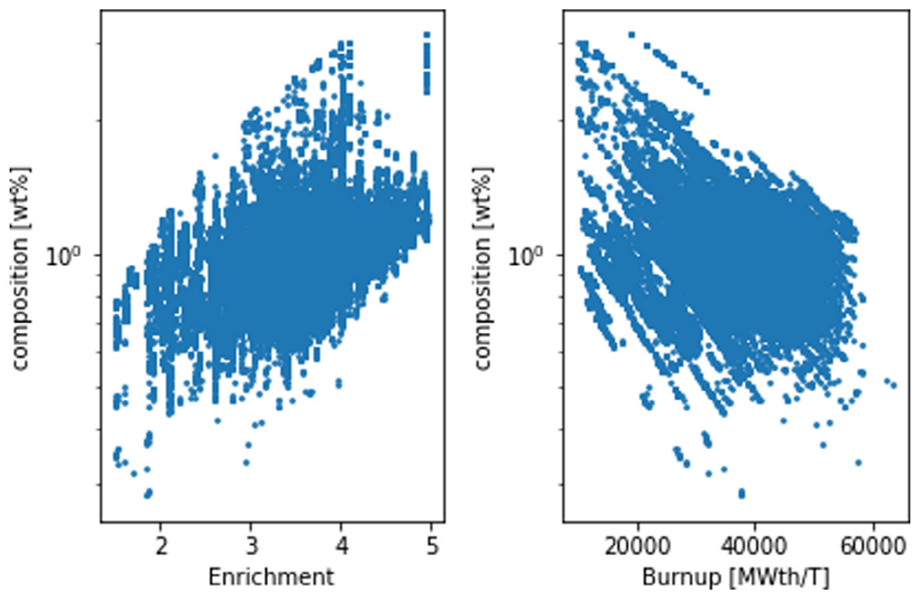

Fig. 5. ^{235}U composition in a UNF assembly is somewhat proportional to both enrichment and burnup, but is difficult to predict using a simple linear regression algorithm.

Table 2
Table of hyperparameters tested for the neural network model. The bold numbers are the values we used for the final model.

Parameter	Values
Number of hidden layers	1, **2**, 3, 4
Nodes per hidden layer	4, 16, 32, 64, **128**
Dropout rate	**0.0**, 0.2, 0.5

with a specific burnup and enrichment. Second, we compared the waste characteristics of an assembly to all assemblies. Third, we compared the predicted total PWR UNF inventory with the UDB. The metric for error is calculated as relative error percentage, ϵ,

$$\epsilon = \frac{x_{data} - x_{model}}{x_{data}} \tag{2}$$

where x_{data} and x_{model} are composition values from the data and model, respectively. This metric provides a fair assessment that is sensitive even to isotopes comprising a small part of the total fuel mass. Notably a large error percentage in the prediction of these trace isotopes reflects a large error with respect to weight %, not necessarily a large absolute mass difference.

We investigated the model accuracy with respect to fuel cycle metrics such as activity and decay heat. The purpose of this investigation is to see how this model can accurately predict UNF inventory profile when used in a NFC simulator.

We compared parameters of the UNF inventory such as activity and decay heat, using the Python toolkit for Nuclear Engineering (PyNE) (Scopatz et al., 2012).

Ideally, the model should be tested against data that is not part of the training data. However, given that the purpose of this model is to allow accessible and quick depletion calculation for fuel cycle

$$\begin{bmatrix} 2.1 & 3.1 & 3.1 & 3.1 & 3.1 & 4.1 & 4.1 & 4.1 & 3.1 \\ 3.1 & 3.1 & 3.1 & 3.1 & 3.1 & 3.1 & 4.1 & 3.5 & 3.2 \\ 3.1 & 3.1 & 3.1 & 3.1 & 3.1 & 3.1 & 3.1 & 3.1 & 3.1 \end{bmatrix}$$

Fig. 6. Enrichment matrix defined for reactor with 3 batches, and 9 assemblies per batch.

$$\begin{bmatrix} 30,000 & 30,000 & 30,000 \\ 32,000 & 34,000 & 40,000 \\ 33000*(1.01)**(t) & 33000*(1.01)**(t) & 33000*(1.01)**(t) \end{bmatrix}$$

Fig. 7. Burnup matrix defined for reactor with 3 batches, and 3 assemblies per batch. The values could also be passed as an equation for time. In this case, the burnup for the last (and equilibrium) increases with time.

simulations, the model would suffice if it predicts the dataset well. In other words, the goal of the model is to be able to reproduce, in a continuous manner, the range of fuel depletion calculations in the database without access to the particular dataset and to have comparable performance to simple interpolation with respect to accuracy. This work demonstrates a general approach for implementing rapid depletion models in fuel cycle simulations. To create depletion models for other reactor designs or depletion parameters, one would simply change the dataset to a set of depletion calculations performed for those specific reactor designs and operational parameters.

2.6. Model implementation in CYCLUS

The trained model is exported to a file that can be plugged into external codes. Since CYCLUS allows the developer to design archetypes in python, we developed a Python-based reactor module that behaves similarly to the CYCAMORE reactor but calculates depleted fuel composition using the imported model instead of a recipe. The user defines a burnup and enrichment matrix for the reactor, and can even vary individual assembly burnups (Figs. 6 and 7). The rows are the number of batches, and the columns are the number of assemblies in a batch.

This reactor module is also available on Github (Bae, 2019).

This sort of implementation can be done with a recipe-based approach for modeling reactor depletion if the user defines multiple output recipes. However, the user can only define the recipe of a batch. Implementing this trained neural network model will allow the user to vary burnup and enrichment for individual assemblies, as well as vary fuel residence time and burnup with reactor lifetime or simulation time. Such capability will be useful in simulating the U.S. UNF inventory in the future, where the burnup of Light Water Reactor (LWR) fuel will increase with advanced fuel technology.

3. Results

The model performed better than using the average recipe in predicting the U.S. UNF, with negligible increase in computational time.

3.1. Depletion calculation time and file size

For 100 random sets of burnup and enrichment depletion predictions, the model takes 0.27 s to output discharge composition, while searching the database for assemblies with the closest burnup and enrichment (using Pandas) takes 21.8 s. Comparatively, 100 ORIGEN calculations take 118 s. Using the model achieves 43,700% reduction in time and does not require libraries, or a reactor physics code. The standalone model pickle file is only 38 Kb, while the curated database (.csv) is 330 Mb.

3.2. Assembly comparison

Ten data points were randomly sampled from the UDB, and were compared with the model predictions to observe two things:

1. What isotopes the model is good/bad at predicting
2. What burnup/enrichment range the model is good/bad at predicting

Figs. 8–11 show that the model generally has a high relative error percentage for ^{226}Ra (average concentration $6.0 \times 10^{-12}\%$), ^{227}Ac (average concentration $2.3 \times 10^{-12}\%$), and curium isotopes. The absolute prediction errors are quite small (averaging $1e-11$), but the large percent errors are due to the small value of the data. There was not a notable difference in the error values for enrichment and burnup variations.

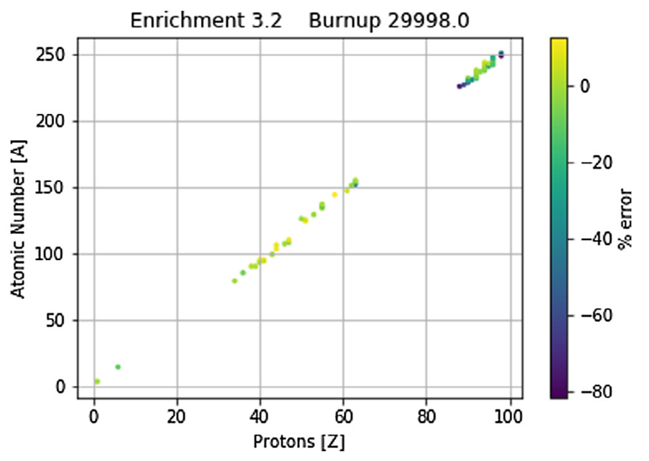

Fig. 8. Isotopic composition prediction error for an assembly with 29.998 $\frac{GWd}{MT}$ burnup and 3.2% enrichment.

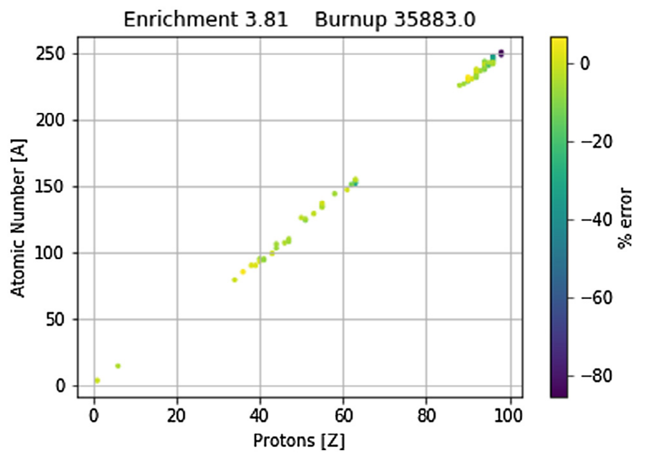

Fig. 9. Isotopic composition prediction error for an assembly with 35.883 $\frac{GWd}{MT}$ burnup and 3.81% enrichment.

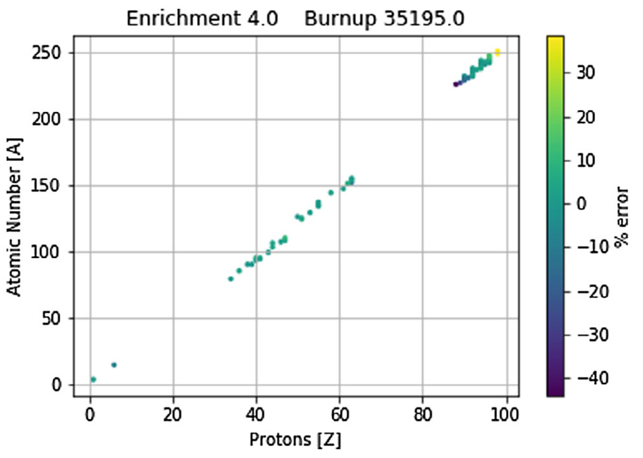

Fig. 10. Isotopic composition prediction error for an assembly with 35.193 $\frac{GWd}{MT}$ burnup and 4.0% enrichment.

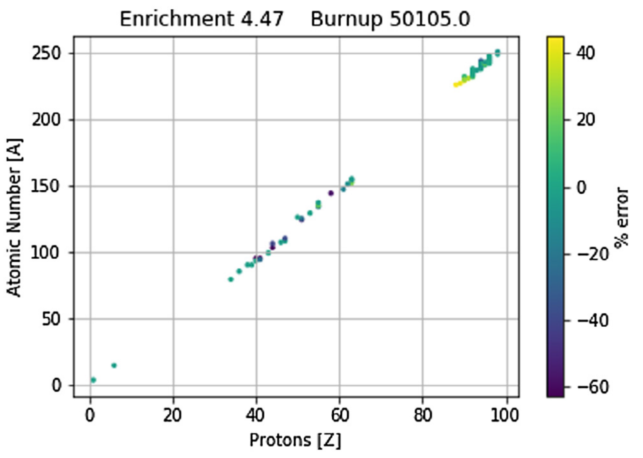

Fig. 11. Isotopic composition prediction error for an assembly with 50.105$GWdMT$ burnup and 4.47% enrichment.

3.3. U.S. UNF inventory comparison

In this section, we compare three UNF inventory composition model approaches. The only difference is the composition of the assemblies. The three different inventory compositions were acquired by:

1. **Data**: directly query the assembly composition from the UDB.
2. **Prediction**: neural network prediction of depleted composition using burnup and enrichment from database
3. **Recipe**: using a single composition (recipe) for all assemblies. Assumes all compositions are the same.

The median values for burnup and initial enrichment are 41,552 MWd/MT and 3.85 (wt%), respectively. The concentrations of major isotopes in the assembly are in Table 3.

We compare the three composition predictions according to:

1. Isotopic inventory
2. Waste metrics (activity and decay heat)
3. Equivalent fissile inventory (equivalent ^{239}Pu)

The UDB contains discharged assembly data from nuclear reactors in the United States up to May of 2013. We added all the UNF assemblies in the database and evaluated the inventory as it was in 2013. Table 4 shows the comparison of the inventories.

3.3.1. Isotopic inventory

In terms of isotopic composition accuracy, the trained neural network model outperforms the mean recipe method for all isotopes. Fig. 12 shows the relative error between the full database, model prediction, and the mean recipe for major isotopes. For plutonium isotopes, the trained neural network model far outperforms the mean database (Fig. 13).

3.3.2. Waste management metrics

The trained neural network excellently predicts the activity and decay heat metrics. Figs. 14 and 15 show the relative error percent of the decay heat and activity predictions per assembly. The model predicts 99.5% of assemblies with an error of less than 1%. Figs. 16 and 17 show the relative error of the decay heat and activity calculated with the average recipe method. Unsurprisingly, the error increases as the actual burnup and enrichments diverge from the average.

Table 4
Comparison of PWR UNF inventory in the U.S, obtained from direct data query, recipe approach, and neural network prediction.

Metric	Data	Recipe	Prediction
^{239}Pu mass [t]	320.37	351.70	321.38
^{137}Cs mass [t]	63.84	66.64	63.73
^{235}U mass [t]	464.60	487.94	474.14
^{238}U mass [t]	42,171	42,016	42,162
Decay Heat [MW]	193.39	198.55	193.33
Activity [$E+21$Bq]	2.79	2.84	2.75

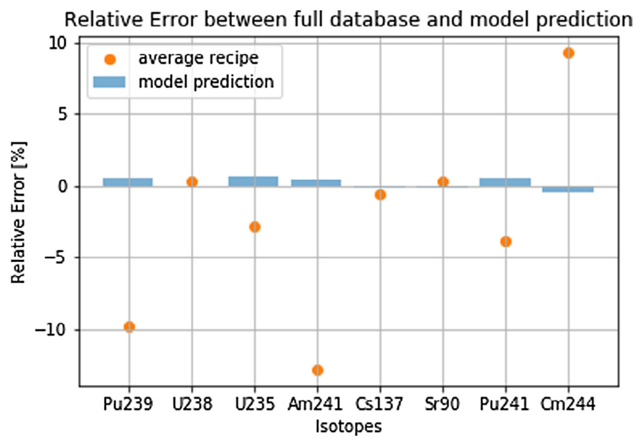

Fig. 12. Neural network model prediction error relative to median UDB recipe, for key isotopes.

Table 3
Isotopic concentration of the assembly with median burnup and enrichment. This composition is used for the recipe method.

Isotopes	^{235}U	^{238}U	^{239}Pu	^{137}Cs	^{90}Sr
Concentration [wt%]	1.076	92.66	0.77	0.14	0.061

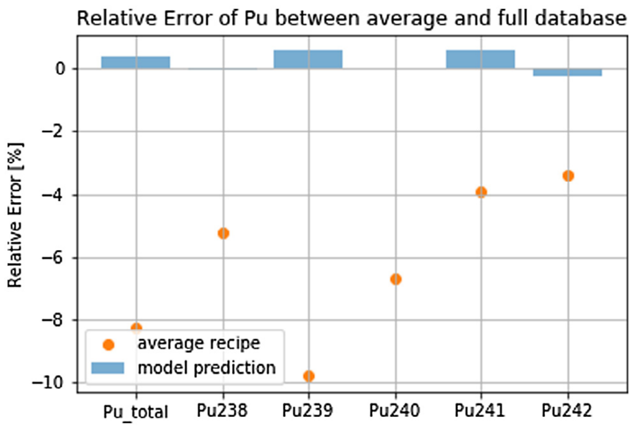

Fig. 13. Neural network model prediction error relative to median UDB recipe, for plutonium isotopes.

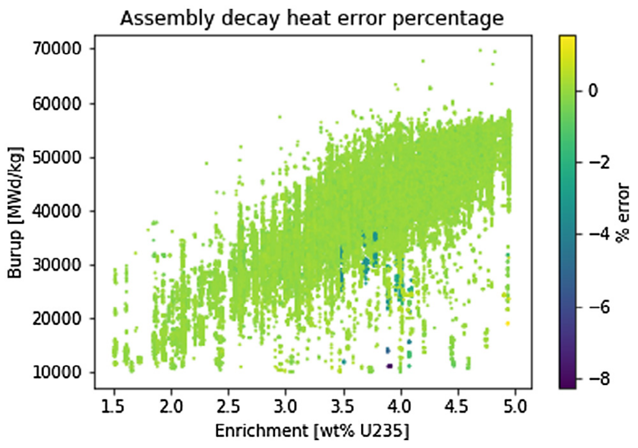

Fig. 14. Relative error percentage for predicting the decay heat of individual assemblies.

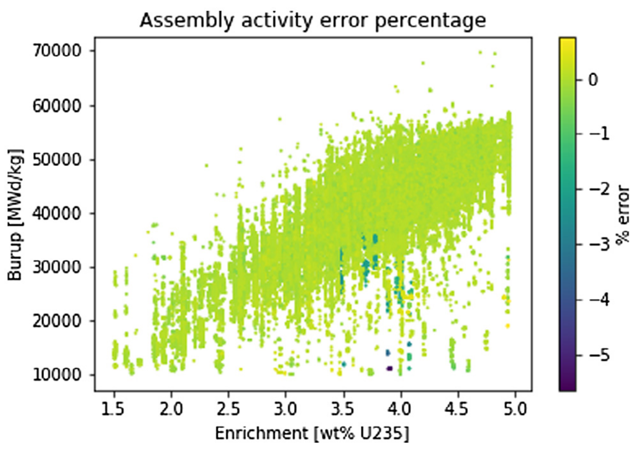

Fig. 15. Relative error percentage for predicting the activity of individual assemblies.

Table 5 shows the decay heat and activity comparison in the years 2020, 2100, and 3100. The total error is less than 1.1% for all metrics at all time periods. Fig. 18 shows relative error in activity and decay heat as a function of time. It shows that the model outperforms the average recipe method in predicting waste metrics.

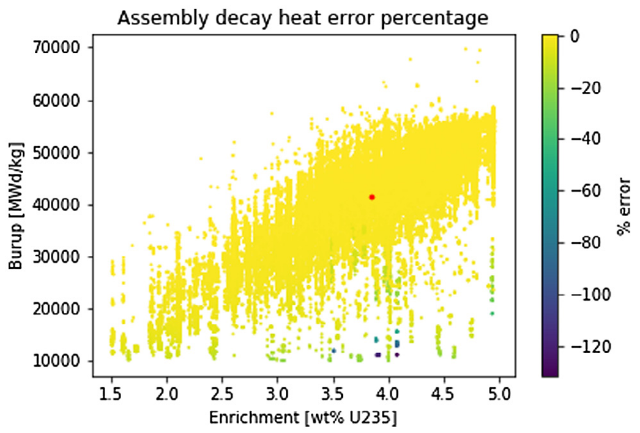

Fig. 16. Relative error in decay heat calculated by the average recipe method. The red point is the median enrichment and burnup. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

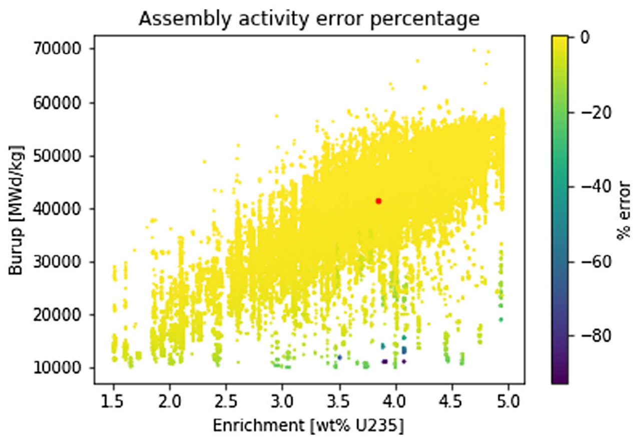

Fig. 17. Relative error in activity calculated by the average recipe method. The red point is the median enrichment and burnup.

3.3.3. *Assembly fissile quality*

Fissile quality is frequently quantified in units of ^{239}Pu equivalent, shown in Table 6 (Anon, 1989). This value is calculated by aggregating the weighted fissile values of each isotope in a material. The ^{239}Pu equivalent factors are different for fast neutron spectrum and thermal neutron spectrum reactors (Baker and Ross, 1963) (factors shown in Table 6). The equivalent fissile value is calculated by:

$$\mathrm{Pu_{eq}} = \sum_i w_i m_i \tag{3}$$

$i \in [^{235}U, ^{238}Pu, ^{239}Pu, ^{240}Pu, ^{241}Pu, ^{242}Pu, ^{242}Am]$
w_i = equivalent weighting factors
m_i = mass of iso i

Where the variables represent the mass of each isotope.

Fig. 19 shows the fast spectrum ^{239}Pu equivalent value of the UNF inventory plotted over time. The trained model outperforms the recipe method. The initial falls for all three lines are due to the decay of plutonium 241, which has a half-life of 14 years.

Table 5
Decay heat and radioactivity values and errors for years 2020, 2100, and 3100.

Metric	Year	UDB [MW]	Prediction [MW]	Error [%]
Decay Heat	2020	40.97	41.07	−0.24
	2100	16.42	16.47	−0.35
	3100	3.13	3.14	−0.15
		UDB [10^{19}Bq]	Prediction [10^{19}Bq]	Error[%]
Activity	2020	46.70	46.60	0.21
	2100	6.39	6.38	0.07
	3100	0.36	0.36	−0.17

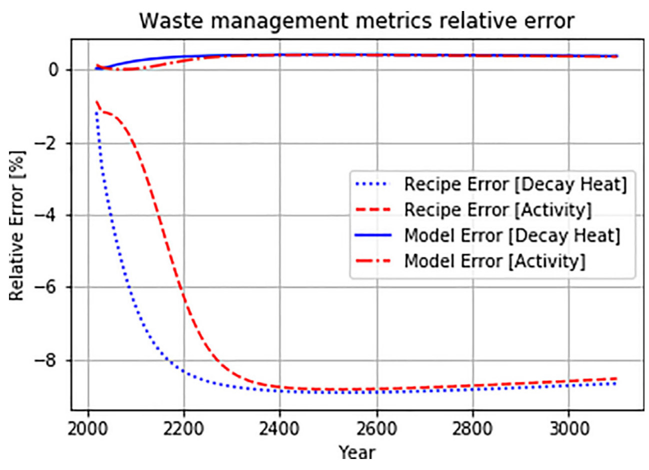

Fig. 18. Relative error of waste management metrics for UNF inventory generated by the average recipe and the prediction model.

Table 6
^{239}Pu equivalence factors from (Anon, 1989). Factors are separately reported for thermal and fast spectra.

	LWR (Thermal)	Fast Breeder Reactor (FBR) (Fast)
^{235}U	+0.8	+0.8
^{238}Pu	−1.0	+0.44
^{239}Pu	+1.0	+1.0
^{240}Pu	−0.4	+0.14
^{241}Pu	+1.3	+1.5
^{242}Pu	−1.4	+0.037
^{241}Am	−2.2	−0.33

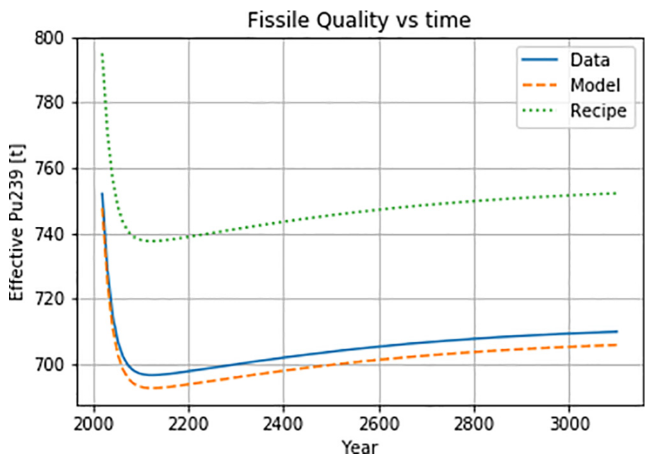

Fig. 19. ^{239}Pu equivalent value in time for three inventories. The model predictions match closely with the value from the database.

3.4. CYCLUS implementation

In this work, we trained a neural network model and implemented it as a CYCLUS reactor agent that predicts UNF composition. The model predicts spent fuel composition based on customizable reactor parameters such as discharge burnup, initial enrichment, cycle time, and power capacity. The created archetype in CYCLUS also allows users to define time-dependent equations instead of constants for reactor parameters. The user can define an enrichment-burnup matrix for each assembly in each batch, and the burnup and enrichment values can be equations in time. This way, users can implement reactor facilities in which the reactor parameters change in time (e.g. to represent reactor uprates, industry burnup trends, etc.).

Figs. 20 and 21 show the discharge fuel composition of a reactor facility in which we increased the discharge burnup from 33,000 to 71,710 MWd/MT over 25 discharge cycles. It should be noted that

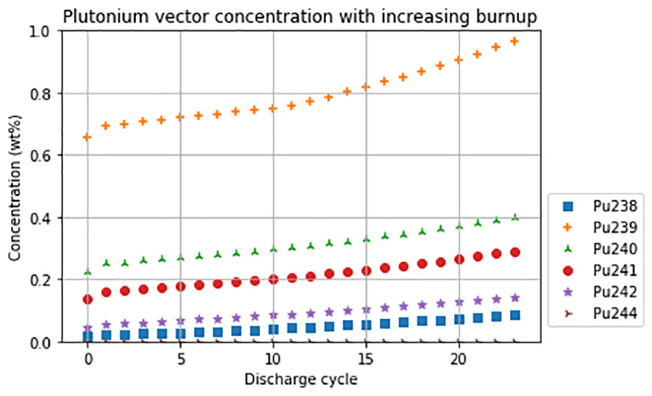

Fig. 20. Plutonium isotope composition in discharge fuel over discharge cycle. The model does not predict well for the target burnup values that are over the burnup listed in the training dataset.

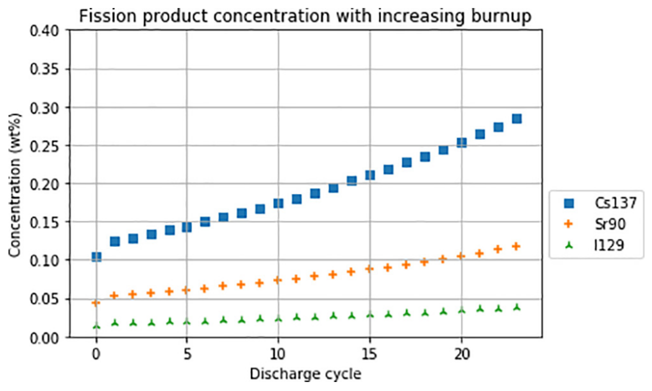

Fig. 21. Fission product concentration in discharge fuel over discharge cycle. Increased discharge burnup leads to higher fission product concentration.

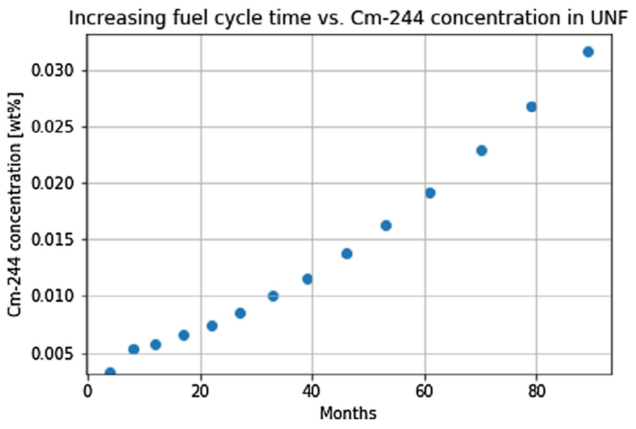

Fig. 22. Discharge and refueling cycles can be defined as an equation of time in this reactor archetype. Discharge burnup is scaled to take into account longer fuel residence time, and leads to increase in discharge fuel ^{244}Cm composition.

the model does not take into account the plausibility of such fuel depletion. For example, it would be nearly impossible for a PWR to burn 2% enriched UOX fuel to 70,000 MWd/MT.

The user can also define time-varying cycle time and refueling time for the reactor model, as shown in Fig. 22.

3.4.1. Applications and use cases

The capability to set dynamic reactor parameters allows simulation of various future transition scenarios that depend on UNF inventory characteristics, such as minor actinide (MA) inventory. Users can simulate future scenarios in which the discharge burnup of reactors increases over time to reveal impacts on MA inventory and, correspondingly, transition speed.

With advances in materials, reactors may have longer cycle times and higher fuel discharge burnups. This dynamic reactor model will be able to account for the changes in these reactor parameters. Also, users can simulate potential power uprates in currently existing fleets, and estimate corresponding impacts on UNF inventory.

4. Conclusion and discussion

This work shows that depleted fuel composition predictions in fuel cycle simulators can be improved by using a predictive neural network model. The neural network model predicted the UNF inventory with less than 5% error for important isotopes, less than 1% error for waste management profile metrics, and less than 0.1% error for ^{239}Pu equivalence. The predictive model outperformed the average recipe method in every metric.

We implemented this model in CYCLUS to provide a reactor model with dynamic reactor parameters, which can simulate potential future improvement scenarios in reactor operation.

This work also shows that open-source depletion models that run quickly can be implemented using prediction models, trained with datasets from complex depletion algorithms. NFC simulators struggle to find a balance between fidelity and rapidity. Using a high-fidelity model is prohibitively computationally expensive since NFC simulators may need to run hundreds of depletion calculations for multiple facilities. On the other hand, using simpler methods like recipes may be too simple for some applications since they do not take into account variations in fuel parameters such as burnup. A well-trained predictive algorithm can find a middle ground between rapidity and fidelity.

Ideally, the model would be validated against an external dataset, instead of the dataset used to train the model. However, the purpose of this work is to create a model that can quickly reproduce the database without having access to the database, which is private data and large in size. The pickled file that contains the model and data scaling objects is only 38.2 kB, meaning that it can be easily distributed and imported in external software, without revealing detailed information about the actual dataset.

An accessible, large-scale depletion database like the UDB are valuable but rare. The value of data is increasing, with the advancement of data science and machine learning. For the long-term advancement of the field, the community should encourage collection and storage of more data and simulation results in a central repository.

Importantly, this neural network is only appropriate for PWR depletion calculations covered by the parametric space of the training set. So, the user will need to remain inside the realm of applicability when using neural networks trained in this way.

Future work

A trained model is only as good as the data it is trained on. This work can be improved and expanded by generating more comprehensive depletion data that covers a wider range of enrichment and burnup ranges. An automation script might run SCALE/ORIGEN to perform depletion calculations for a wide range of enrichment (e.g. 0.7–4.99 wt%) and burnup (e.g. 0–80,000 MWd/MT), for a single assembly design. Assumptions of criticality and irradiation time should be made as well. The results could then be parsed into a Comma-separated values (CSV) file and stored for the training of a new model. This will allow better prediction of the model for higher burnups and 'fringe' burnup-enrichment assemblies.

Methods used in this paper have the potential to be expanded into more complicated problems. An interesting application of this method is for Molten Salt Reactor (MSR) system optimization. Current work on MSRs include optimization non-core operating parameters such as reprocessing scheme and flow rate. Fuel transmutation calculations can be usually computationally burdensome for MSR simulations, since the flowing fuel is depleted and reprocessed continuously. MSR models implement semi-continuous methods in which the depletion-to-reprocessing time is very short (usually 3 days), which makes an MSR lifetime simulation (if 60 years) require $\sim 7,300$ depletion calculations. The computational burden makes it impossible to use brute-force methods, such as grid search of all possible parameters. However, if a quick depletion calculation becomes possible with a well-trained prediction model, the computational burden will dramatically decrease. However, problems with generating enough training data, accuracy, and the model's ability to extrapolate remain.

Acknowledgments

The authors thank Dr. Kaushik Banerjee for providing the data with the composition of depleted assemblies.

This research was performed using funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Program (Project 16–10512, DE-NE0008567) 'Demand-Driven Cycamore Archetypes'.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.anucene.2019.107230.

References

Anon, 1989. Plutonium fuel an assessment. Report by an expert group, Tech. Rep. INIS-XN–254, Nuclear Energy Agency.http://inis.iaea.org/Search/search.aspx?orig_q=RN:21076679..

Bae, J.W., 2019. jbae11/ann_pwr: Zenodo Release.https://doi.org/10.5281/zenodo.3353745.https://zenodo.org/record/3353745#.XT4iKSYpBhE..

Bae, J.W., Chee, G., Huff, K., Rykhlevskii, A., 2019. jbae11/depletion_rom: Zenodo Release.https://doi.org/10.5281/zenodo.3353747.https://zenodo.org/record/3353747#.XT4iRyYpBhE..

Baker, A., Ross, R., 1963. Comparison of the value of plutonium and uranium isotopes in fast reactors, USAEC, Breeding, Economics, and Safety in Large Fast Breeder Reactors (Argonne National Laboratory, Illinois, 1963), pp. 329–365..

Bowman, S.M., 2011. SCALE 6: comprehensive nuclear safety analysis code system. Nucl. Technol. 174 (2), 126–148. https://doi.org/10.13182/NT10-163. URL: http://epubs.ans.org/?a=11717.

Chollet, F., et al., 2015. Keras,https://keras.io..

Coquelet-Pascal, C., Tiphine, M., Krivtchik, G., Freynet, D., Cany, C., Eschbach, R., Chabert, C., 2015. COSI6: a tool for nuclear transition scenario studies and application to SFR deployment scenarios with minor actinide transmutation. Nucl. Technol. 192 (2), 91–110. https://doi.org/10.13182/NT15-20. URL: https://www.tandfonline.com/doi/full/10.13182/NT15-20.

Croff, A.G., 1980. User's manual for the ORIGEN2 computer code, Tech. Rep. ORNL/TM-7175. Oak Ridge National Lab..

Feng, B., Dixon, B., Sunny, E., Cuadra, A., Jacobson, J., Brown, N.R., Powers, J., Worrall, A., Passerini, S., Gregg, R., 2016. Standardized verification of fuel cycle modeling. Ann. Nucl. Energy 94, 300–312. https://doi.org/10.1016/j.anucene.2016.03.002. URL: http://www.sciencedirect.com/science/article/pii/S0306454916301098.

Flanagan, R., 2014. Bright-lite,https://github.com/bright-dev/bright-lite.

Gidden, M.J., Wilson, P.P.H., 2016. A methodology for determining the dynamic exchange of resources in nuclear fuel cycle simulation. Nucl. Eng. Des. 310, 378–394. https://doi.org/10.1016/j.nucengdes.2016.10.029. URL: https://www.sciencedirect.com/science/article/pii/S0029549316304101.

Gregg, R., Grove, C., 2012. In: Proc of the IChemE nuclear fuel cycle conference, Manchester, United Kingdom, Manchester, United Kingdom.http://www.icheme.org/events/conferences/nuclear..

Huff, K.D., Fratoni, M., Greenberg, H., 2014. Extensions to the cyclus ecosystem in support of market-driven transition capability. In: Transactions of the American Nuclear Society, Fuel Cycle Options Analysis – III. American Nuclear Society, Anaheim, CA, United States, pp. 245–248. lLNL-PROC-656426.

Huff, K.D., Gidden, M.J., Carlsen, R.W., Flanagan, R.R., McGarry, M.B., Opotowsky, A. C., Schneider, E.A., Scopatz, A.M., Wilson, P.P.H., 2016. Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework. Adv. Eng. Softw. 94, 46–59. https://doi.org/10.1016/j.advengsoft.2016.01.014. arXiv: 1509.03604.http://www.sciencedirect.com/science/article/pii/S0965997816300229.

Jacobson, J., Yacout, A., Matthern, G., Piet, S., Shropshire, D., Jeffers, R., Schweitzer, T., 2010. Verifiable fuel cycle simulation model (VISION): a tool for analyzing nuclear fuel cycle futures. Nucl. Technol. 172 (2), 157–178.

Leniau, B., Mouginot, B., Thiolliere, N., Doligez, X., Bidaud, A., Courtin, F., Ernoult, M., David, S., 2015. A neural network approach for burn-up calculation and its application to the dynamic fuel cycle code CLASS 81, pp. 125–133.https://doi.org/10.1016/j.anucene.2015.03.035.http://www.sciencedirect.com/science/article/pii/S0306454915001693..

Leppanen, J., Pusa, M., Viitanen, T., Valtavirta, V., Kaltiaisenaho, T., 2015. The serpent Monte Carlo code: status, development and applications in 2013. Ann. Nucl. Energy 82, 142–150. https://doi.org/10.1016/j.anucene.2014.08.024. URL: http://www.sciencedirect.com/science/article/pii/S0306454914004095.

McKinney, W., 2010. Data structures for statistical computing in python. In: van der Walt, S., Millman, J. (Eds.), Proceedings of the 9th Python in Science Conference, pp. 51–56.

Mouginot, B., Clavel, J.B., Thiolliere, N., 2012. CLASS, a new tool for nuclear scenarios: description & first application, world academy of science, engineering and technology. Int. J. Math., Computat., Phys., Electr. Comput. Eng. 6 (3), 232–235. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.5389&rep=rep1&type=pdf.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Peterson, J., Scaglione, J., 2015. Fuel Cycle Applications with the Data Stored Within the Unified Database, 5..

Peterson, J., Lefebvre, R., Smith, H., Ilas, D., Robb, K., Michener, T., Adkins, H., Scaglione, J., 2013. Used Nuclear Fuel Storage Transportation & Disposal Analysis Resource and Data System (UNF-ST&DARDS), Tech. rep., FCRD-NFST-2013-000117 Rev. 0, US Department of Energy Nuclear Fuels Storage and Transportation Planning Project..

Peterson, J., Olson, C., Aubin, J.St., Craig, B., 2017. Additional capability being developed from UNF-ST&DARDS unified database: system analysis and fuel cycle option analysis. Nucl. Technol. 199 (3), 320–329. https://doi.org/10.1080/00295450.2017.1354551. URL: https://www.tandfonline.com/doi/full/10.1080/00295450.2017.1354551.

Scopatz, A., Schneider, E., Biegalski, S., Landsberger, S., Deinert, M., Yim, M.-S., 2011. Essential Physics for Fuel Cycle Modeling & Analysis 269..

Scopatz, A., Romano, P.K., Wilson, P.P.H., Huff, K.D., 2012. PyNE: Python for Nuclear Engineering. In: Transactions of the American Nuclear Society, vol. 107. American Nuclear Society, San Diego, CA, USA.

Skutnik, S.E., Sly, N.C., Littell, J.L., 2016. CyBORG: an ORIGEN-based reactor analysis capability for cyclus. In: Transactions of the American Nuclear Society. Fuel Cycle and Waste Management: General-II, vol. 115. American Nuclear Society, Las Vegas, Nevada, United States, pp. 299–301.

Stone, M., 1974. Cross-validatory choice and assessment of statistical predictions. J. R. Stat. Soc. Series B (Methodological), 111–147.

Sunny, E.E., Worrall, A., Peterson, J.L., Powers, J.J., Gehin, J.C., Gregg, R., 2015. Transition Analysis of Promising US Future Fuel Cycles Using ORION. Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States). Tech. rep.,http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/47/088/47088986.pdf.

Yacout, A.M., Jacobson, J.J., Matthern, G.E., Piet, S.J., Moisseytsev, A., 2005. Modeling the Nuclear Fuel Cycle. In: The 23rd International Conference of the System Dynamics Society.

Yacout, A.M., Jacobson, J.J., Matthern, G.E., Piet, S.J., Shropshire, D.E., Laws, C., 2006. VISION – verifiable fuel cycle simulation of nuclear fuel cycle dynamics. Waste Management Symposium. URL: http://www.inl.gov/technicalpublications/Documents/3394908.pdf.