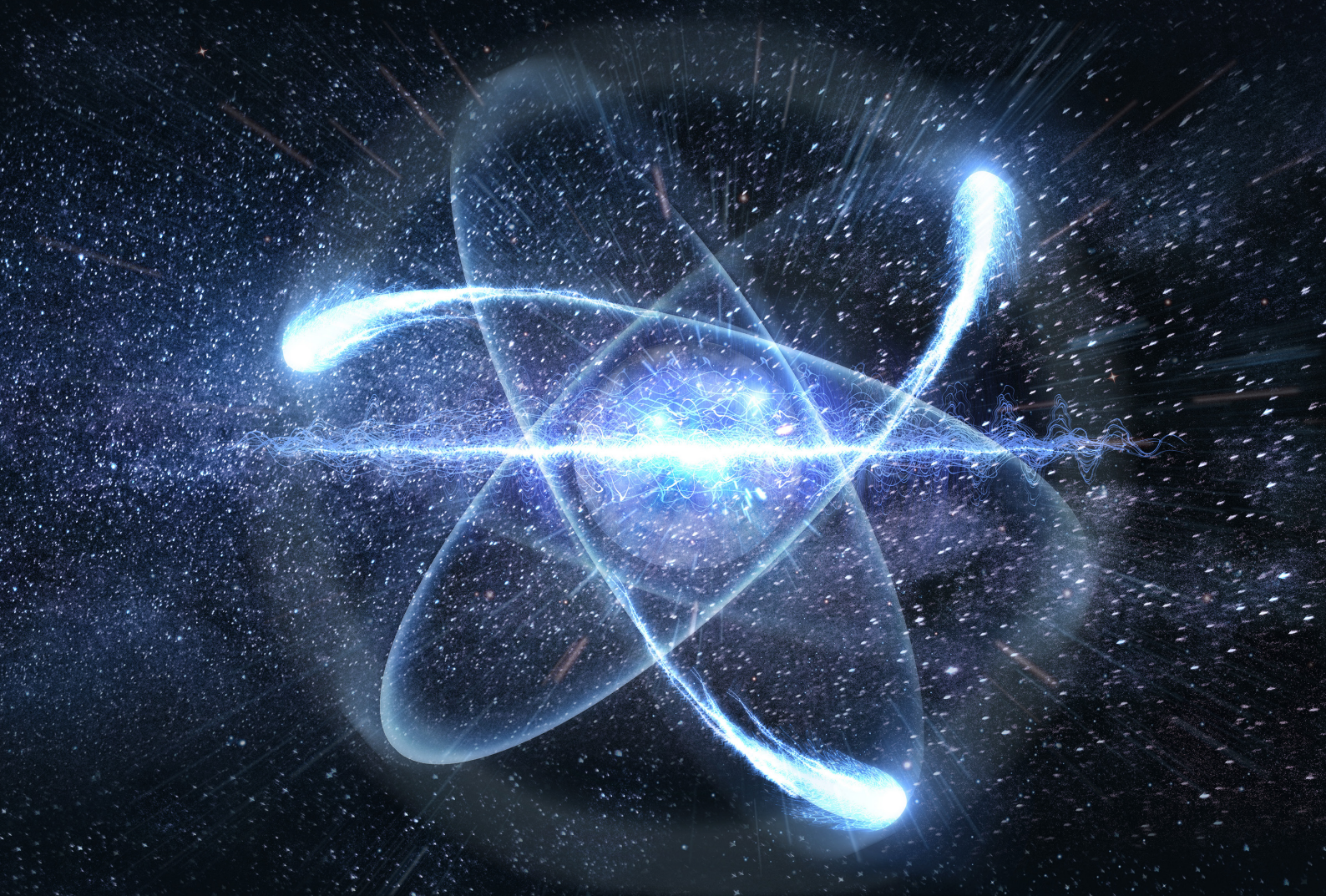
ISSN: 1977-5296

ESARDA

European Safeguards Research & Development Association

The International Journal of Nuclear Safeguards and Non-Proliferation

Issue on Data Analytics for Safeguards and Non-Proliferation



December 2021 Bulletin No.63

ISSN 1977-5296
KJ-BB-21-002-EN-N
doi:10.2760/577687

Number 63
December 2021

Guest Editor
Zoe Gastelum

Editors
Elena Stringa and Andrea De Luca

European Commission, Joint Research Centre,
Directorate G - Nuclear Safety and Security
Nuclear Security Unit G.II.7
T.P. 800, I-21027 Ispra (VA), Italy
Tel. +39 0332-786182
EC-ESARDA-BULLETIN@ec.europa.eu

ESARDA is an association formed to advance and harmonize research and development for safeguards. More information can be found at the following address:

https://esarda.jrc.ec.europa.eu/

Cover designed by Christopher Craig Havenga, (consultant at EC, JRC, G.II.7, Italy)

ESARDA
European Safeguards Research & Development Association

Bulletin

Contents Issue n° 63

Editorial

Peer Reviewed Articles

Editorial

Zoe Gastelum

Dear ESARDA Bulletin readers,

I am honored to be writing to you as the Guest Editor for this special topical issue of the ESARDA Bulletin on Data Analytics for International Nuclear Safeguards and Non-Proliferation.

The human brain has incredible propensity to recognize patterns, identify differences between two scenes, track moving objects through space, summarize themes or ideas, and generalize knowledge after observing few examples. For example, even very young children can recognize a live giraffe at a zoo, a stuffed animal of a giraffe, and a cartoon drawing of a giraffe from only having seen an image in a book. Our brains are constantly analyzing the massive amounts of data that they collect. As more and more data become available to us, as is the case in nuclear safeguards and non-proliferation, we are less able to process it and complete the functions described above. Our working memory is simply too limited.

Data analytics and business intelligence help us to summarize data into human-interpretable formats that we can better understand and act upon. New methods are emerging that further remove humans from the processing of the data – with deep learning, unsupervised approaches, and data tensors hundreds of vectors large. In this special issue, we explore the potential impacts of data analytics methods on nuclear safeguards and non-proliferation.

In May 2019, over 200 people participated in a World Café[1] exercise at the ESARDA Symposium held in Stresa, Italy. In the World Café, participants brainstormed their wishes, challenges, and actions regarding ten topics that had been defined by the ESARDA Reflections Group Report in 2019. Two of the topics addressed in the World Café were directly related to data analytics and have direct relevance to this special issue.

The first data analytics topic was Remote Data Transfer and Artificial Intelligence (AI). Remote data transfer and AI was included for its potential to optimize inspection resources by developing and implementing the remote transfer of data combined with machine learning and artificial intelligence techniques. The wishes and challenges are too numerous to fully describe here, but there were several actions defined that seem to call directly for an issue of the Bulletin such as this one. I have paraphrased them below, with commentary in italics about the role of this special issue in meeting these calls to action:

- Adapt tools that have been developed and proven in other domains to nuclear safeguards and non-proliferation-relevant datasets. *In this special issue, we the approaches described have all been demonstrated, at least in part, in other domains and have been combined or tailored specifically for the specific safeguards and non-proliferation data challenges here.*
- Initiate research projects that prove the feasibility of data analytics techniques for safeguards and non-proliferation, without the requirement to prove generalizability for the full spectrum of safeguards problems. *The four research papers included in this special issue each demonstrates capability for a specific safeguards and non-proliferation challenge. We can anticipate that future solutions will proceed in similar ways, providing proof-of-concept feasibility for specific challenges with customized analytical approaches.*
- Inform and educate [the nuclear non-proliferation and safeguards research community and our stakeholders about the state-of-research, results, and implications] to overcome reluctance. *It is my intention with this special issue, as well as future publications within the Bulletin that will focus on data analytics, that we can more broadly reach stakeholders in the operational sectors of nuclear nonproliferation and safeguards, including the International Atomic Energy Agency, Euratom, and others working at the regional and national levels to share capabilities in this areas specifically focused on nuclear nonprolfieration and safeguards, including the challenges, opportunities, data, etc.*

1 The World Café report is available here: https://esarda.jrc.ec.europa.eu/world-cafe-report-2019_en

The second topic related to data analytics was business intelligence. Business intelligence was identified by the World Café in its opportunity for "mapping, storing, presenting [and] analysing safeguards-relevant data" via interconnected safeguards databases and the use of geographic information systems and visualization tools. Like the remote data analysis and AI topic, the business intelligence topics contained several actions highly relevant for this special issue, again paraphrased with ties to this special issue detailed in italics:

- Make the user community and other stakeholders aware of what business intelligence tools and capabilities are available, and what business intelligence might enable for their needs. *Similar to the bullet under Remote Data Transfer and AI on engagement and knowledge sharing, this special issue is intended to meet this need. Though there are ESARDA members working in this area, we did not receive any full manuscripts related to business intelligence and hope that this is a topic that will be covered in future Bulletin issues, as well as continued publications in other forums including the ESARDA Symposium Proceedings and shared via ESARDA Working Group meetings.*
- Engage and attract experts from the data analytics community in nuclear non-proliferation and safeguards in ESARDA activities, via events and publications. *We also see authors from the data analytics community publishing here in ESARDA for the first time, in cooperation with safeguards authors.*

In this issue, we have representation from three exciting topic areas within the domain of data analytics for nuclear non-proliferation and safeguards. First, we have a research paper from the SCK-CEN Belgian Nuclear Research Centre on the use of machine learning to predict isotopics, burn up, and cooling time of spent nuclear fuel which could significantly improve how spent fuel is verified in cooling ponds. Then, we transition to multi-model machine learning approaches from Sandia National Laboratories and Pacific Northwest National Laboratory to detect nuclear material diversion in reprocessing facilities – a methodology that will potentially decrease the need for expensive and time-consuming destructive assay. Then, we transition to two papers that use natural language process (NLP) to support nuclear non-proliferation analysis. In the first, from Pacific Northwest National Laboratory and North Carolina State University, researchers compared methods of pre-training and fine-tuning language models in order to classify scientific publications using both a binary classifier to determine relevance to the nuclear fuel cycle, and a multi-class classifier to associate the papers with stages of the nuclear fuel cycle, which has the potential to enhance how analysts prioritize and triage publication review. In a second NLP paper from Pacific Northwest National Laboratory, the authors developed a question-and-answer system that was calibrated to nuclear nonproliferation topics and introduced a novel methodology for auditing question-and-answer capability that could enhance how nonproliferation analysts interact with large databases of stored historical data.

A friend of mine recently wrote a piece regarding the use of data analytics to support defense missions, in which she said "You go to war with the data you have," making a play on words of former U.S. Secretary of Defense Donald Rumsfeld's saying "You go to war with the troops you have." While our domain is certainly far from war, nuclear safeguards and non-proliferation verification represents a prominent task impacting international security. I wonder if we might say "In nuclear safeguards, you go on inspection with the data you have." While collection of more data is possible under safeguards agreements – letter are written, follow-up visits are taken, additional overhead imagery is purchased and analyzed, etc. – fundamentally, the types of data that we have available are not changing. They may be increasing in number as more significant quantities of nuclear material are added to safeguards over time, and they may become more frequent as states and operators recognize the potential for sharing remote monitoring data with the IAEA as an avenue to decrease in-person inspections. In the absence of significant budget increases or fundamental changes in scope regarding how safeguards are verified, we need to better exploit our existing data using data analytics and business intelligence methods to answer the recurring calls for increased effectiveness and efficiency.

I thank Bulletin Editors for the opportunity to guest edit this special issue on data analytics – an opportunity that I have long desired and was especially enthusiastic to undertake now that the Bulletin is indexed in SCOPUS. I also thank the members of the ESARDA Verification Technologies and Methodologies especially, for their contributions, for their reviews, and for their support of this activity. And deepest thanks to the authors, peer reviewers, and Editors team for the significant effort that goes into all Bulletin issues.

Sincerely,

Zoe Gastelum

Chair, ESARDA Verification Technologies
and Methodologies Working Group
zgastel@sandia.gov

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

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

Abstract

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

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

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

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

1. Introduction

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

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

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

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

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

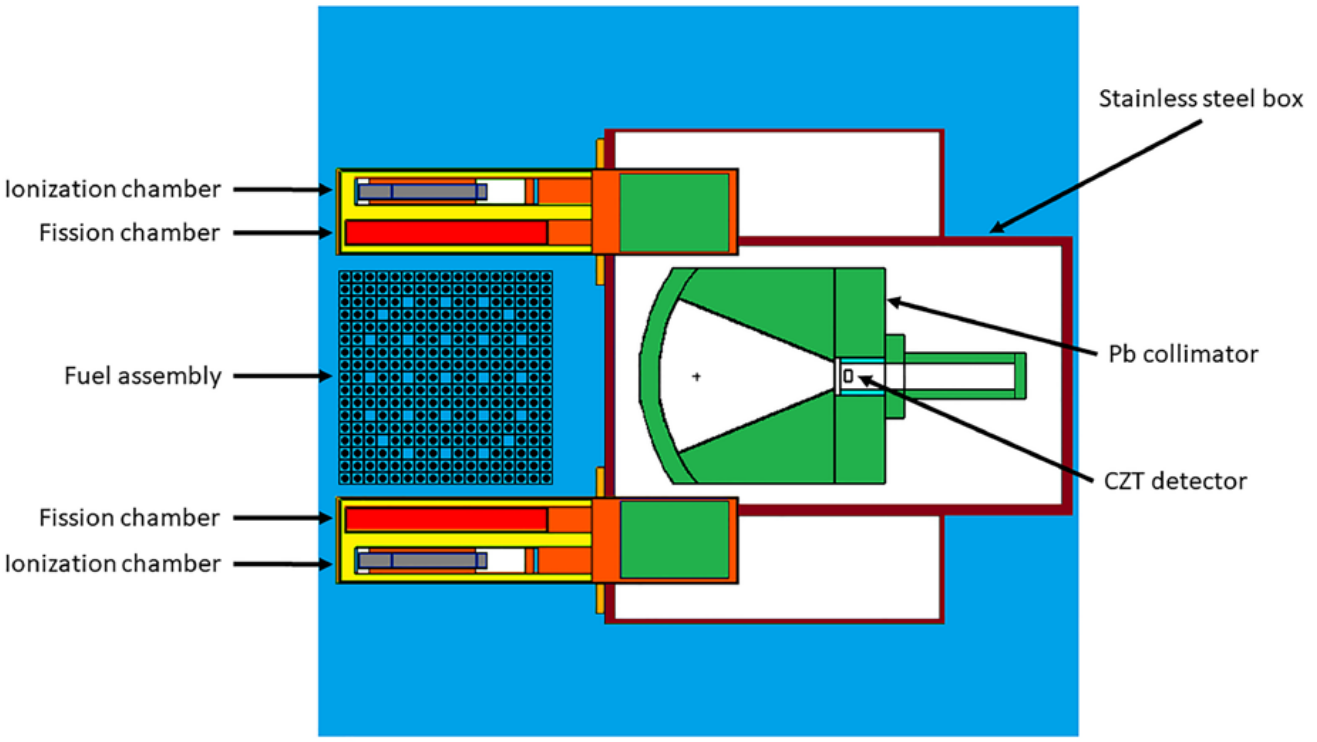


Figure 1: 2-D view of the Monte Carlo model of the Forkball detector measuring a spent fuel assembly. Figure taken from [6].

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

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

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

2. Dataset

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

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

$$CT' = \ln(cooling\ time) + 10 \quad (1)$$

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

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

3. Artificial neural networks

3.1 Basic principles

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

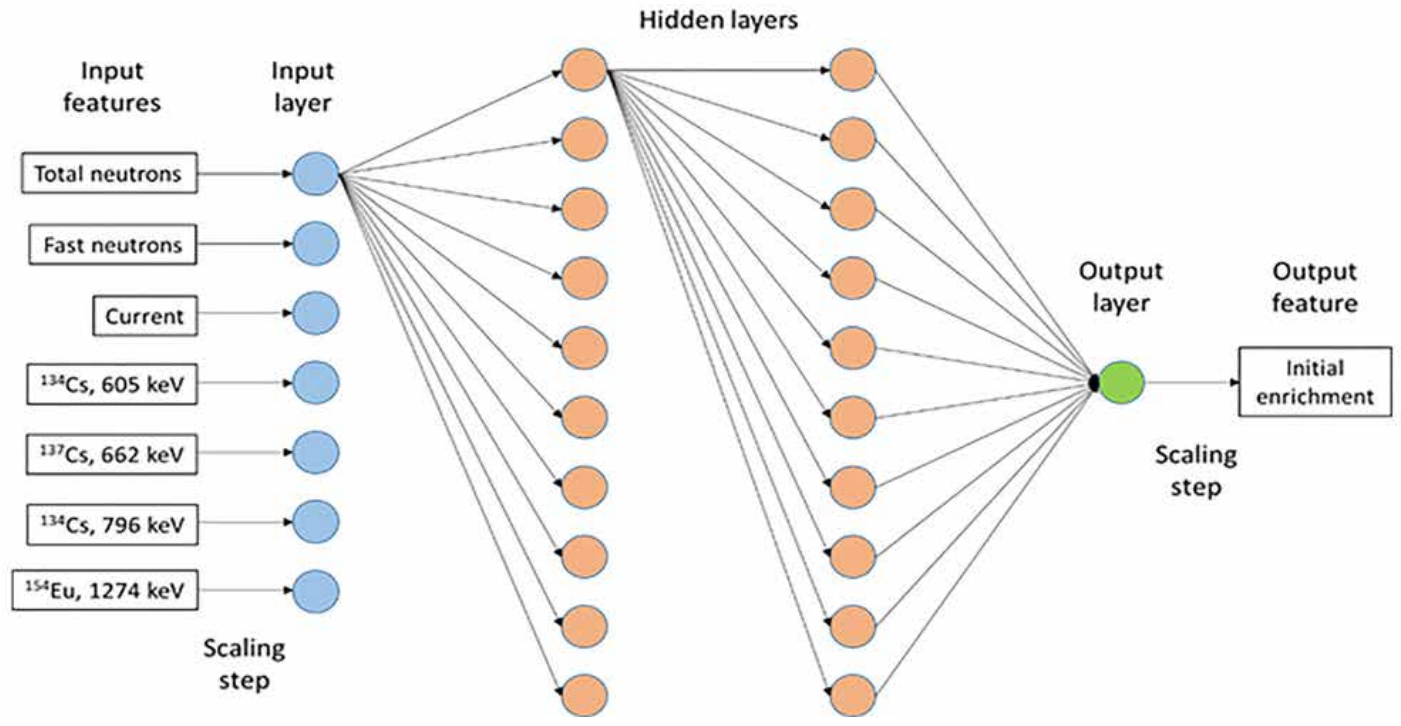


Figure 2: example of ANN architecture used for this study. The ANN includes all input features, two hidden layers each with 10 neurons, and initial enrichment as output variable. The ANN is fully connected but for graphical reasons not all connections among neurons are included in the figure.

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

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

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

The general equation for an artificial neuron is:

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

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

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

3.2 Network architecture

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

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

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

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

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

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

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

4. Results

4.1 One hidden layer ANN

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

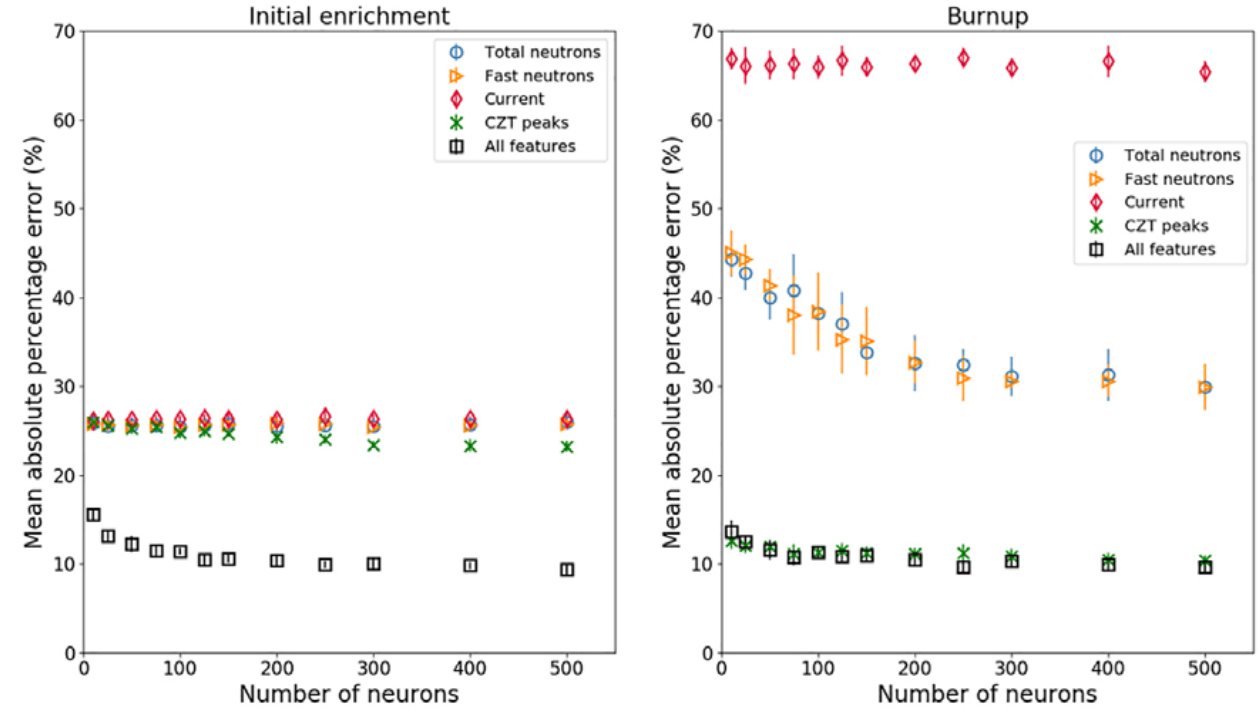


Figure 3: Mean absolute percentage error for the determination of the initial enrichment (left) and burnup (right). The values refer to ANNs with one hidden layer and the mape is shown as a function of the number of neurons in the ANN.

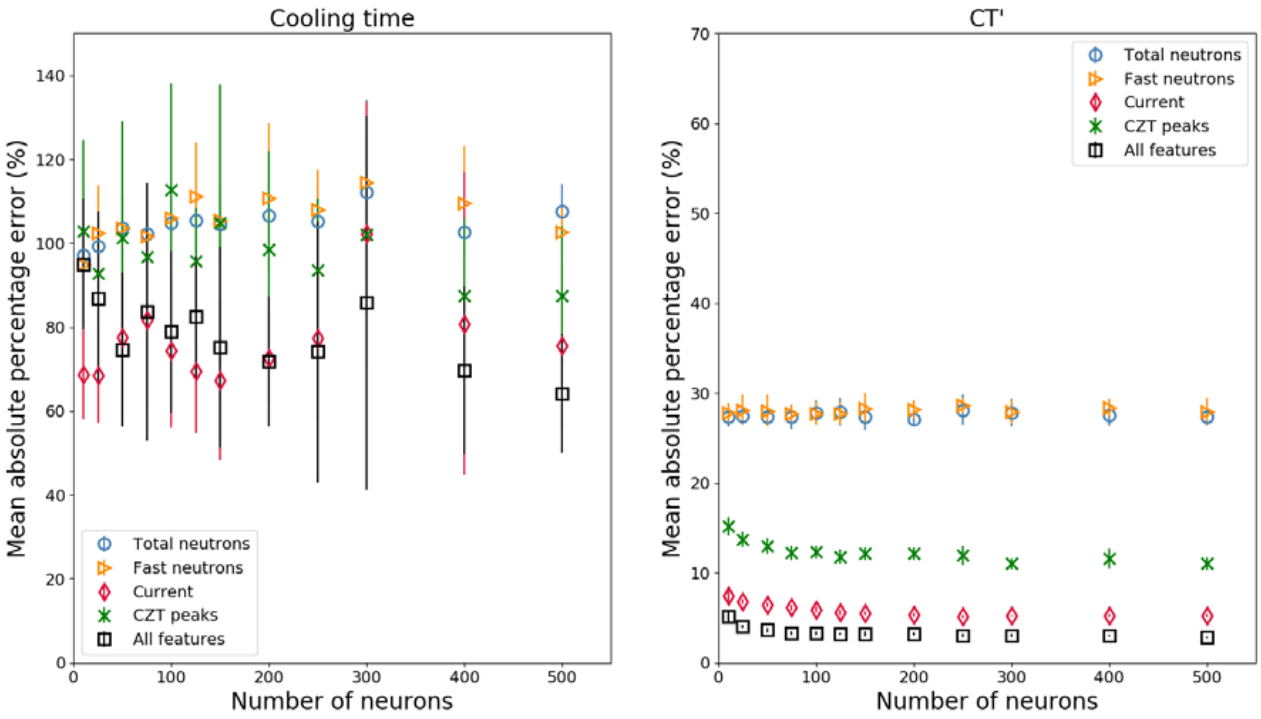


Figure 4: Mean absolute percentage error for the determination of the cooling time (left) and CT' (right). The values refer to ANNs with one hidden layer and the mape is shown as a function of the number of neurons in the ANN.

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

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

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

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

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

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

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

4.2 Two hidden layers ANN

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

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

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

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

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

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

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

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

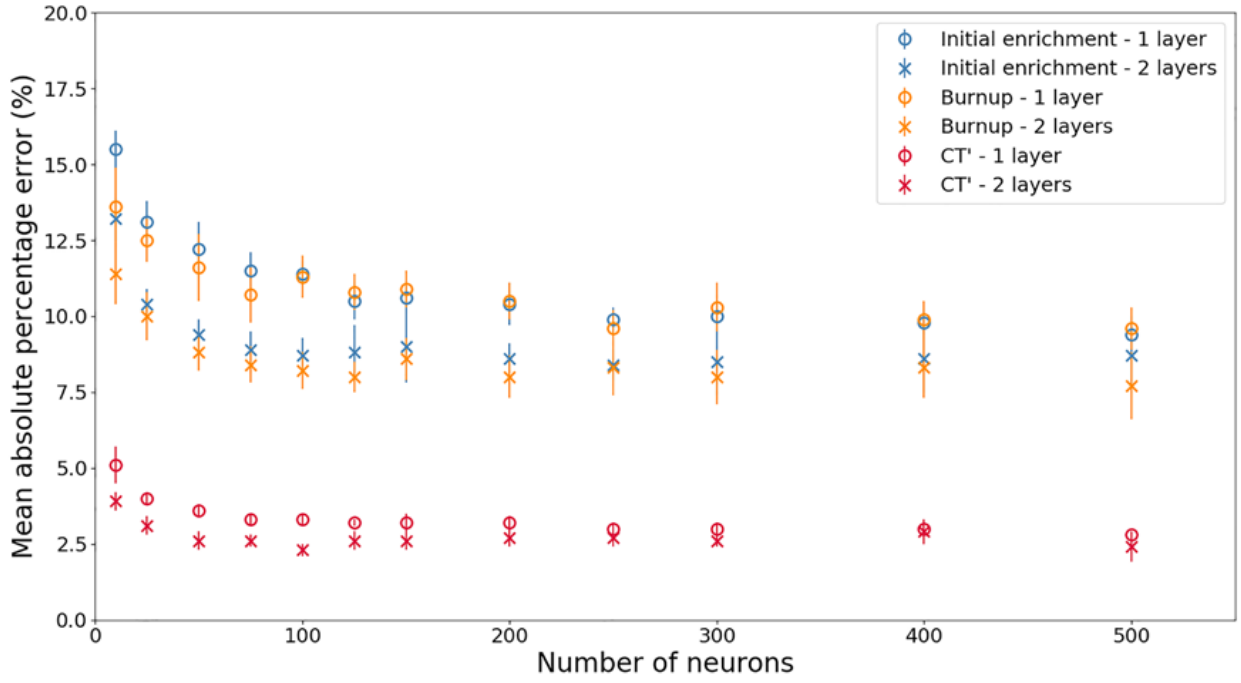


Figure 5: Mean absolute percentage error for the determination of initial enrichment, burnup, and CT'. The values refer to ANNs with one hidden layer and with two hidden layers and are shown as a function of the number of neurons in the ANN.

5. Discussion

5.1 Comparison between one hidden layer and two hidden layers ANN

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

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

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

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

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

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

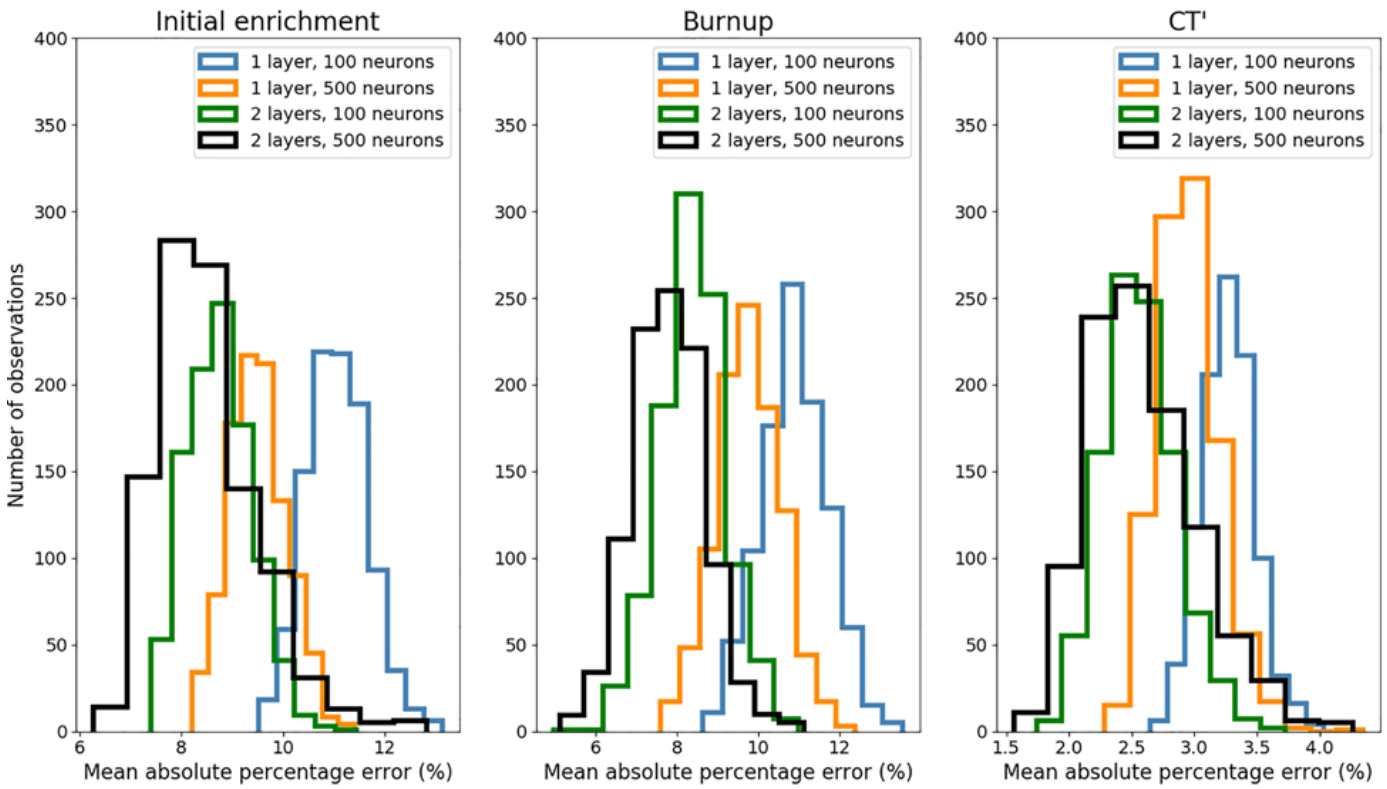


Figure 6: distribution of mean absolute percentage error for the determination of initial enrichment, burnup, and CT'. The training and testing of each ANN architecture was repeated for 1000 iterations, each time with a random partition of the dataset.

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

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

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

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

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

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

from verification campaigns using the Fork detector showed that the relative standard deviation between measured and calculated count rates is less than 8% for neutron detectors and less than 7% for gamma-ray detectors [34].

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

5.2 Optimization of ANN size to reduce overfitting

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

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

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

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

6. Conclusions

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

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

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

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

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

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

7. Legal matters

7.1 Privacy regulations and protection of personal data

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

7.2 Copyright

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

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

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

8. References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Applied Machine Learning for Simulated Reprocessing Safeguards: Unsupervised Networks

Nathan Shoman[1], Benjamin Cipiti[1], Thomas Grimes[1], Ben Wilson[2], Randall Gladen[2]

[1]Sandia National Laboratories
[2]Pacific Northwest National Laboratory
E-mail: nshoman@sandia.gov

Abstract

A goal of the International Atomic Energy Agency (IAEA) is to deter the spread of nuclear weapons through detection of nuclear material and technology misuse. Detecting diversion of nuclear material from large bulk handling facilities, such as a reprocessing plant, is a goal that can prove to be both challenging and resource intensive as it often requires destructive analysis of numerous samples taken from various locations across the facility. The IAEA has sought out methods to develop an integrated system of instrumentation and data processing to reduce this burden. The goal of this work is to leverage machine learning (ML) methods to improve the effectiveness and efficiency of safeguards by utilizing higher uncertainty measurements, such as process monitoring and Non-Destructive Assay measurements, which are not extensively used in traditional safeguards methods. This work is part of a series of two documents that consider the use of ML to improve one aspect of safeguards, namely nuclear material accountancy. This part considers unsupervised networks that are used to detect anomalous behavior that could be indicative of material loss. The unsupervised approach is shown to exceed traditional methodologies but only after several practical barriers have been accounted for and resolved.

Keywords: safeguards; data science; machine learning; nuclear material accountancy; reprocessing

1. Introduction

The International Atomic Energy Agency (IAEA) was established as an organization within the United Nations to promote the peaceful use of nuclear power [1]. One function of the IAEA is the implementation of safeguards for member states. The goal of safeguards is the timely detection of diversion of significant quantities (SQs) of nuclear material for weapon purposes and deterrence of such diversion by the risk of detection. Nuclear material accountancy (NMA) is one method used by the IAEA to implement safeguards. NMA can be thought of as an audit of nuclear facilities that verifies reported quantities of material to ensure they have not been diverted. This is accomplished through several methods such as sampling and process monitoring. Safeguards can be further complemented by other systems such as containment and surveillance (C/S), particularly for large throughput facilities.

Existing NMA systems are well understood and have been implemented at numerous facilities. However, NMA often requires low uncertainty destructive assay (DA) measurements to reach timeliness goals. These measurements are often time consuming and expensive as they must be performed in an analytical laboratory. Other types of measurements, such as process monitoring (PM) and non-destructive assay (NDA), can be used for remote monitoring to lead to lower costs, but often have relatively high uncertainties. Machine learning (ML) has revolutionized many fields and offers promise in safeguards related tasks like anomaly detection. This work hypothesizes that ML could more effectively leverage underutilized measurements with higher uncertainties (e.g. NDA and PM) to improve costs associated with NMA.

2. Background

International safeguards are implemented to guard against diversion of significant quantities of nuclear material. This is defined by the IAEA as the approximate amount of nuclear material for which the possibility of manufacturing a nuclear explosive device cannot be excluded, which for plutonium is 8 kg [2]. One simple approach for the NMA component of international safeguards is item counting. Here, simple counting of discrete items is used to account for items that contain nuclear materials (e.g. fuel assemblies). When

combined with statistics and random sampling, item accounting is indeed the preferred method for facilities where material is most often found in discrete items. However, the focus of this work is large facilities where material is often in bulk form (e.g. powders or solutions) that require methods beyond simple item accounting [3]. The goal of this work is to develop machine learning approaches to improve material accountancy of these large facilities. It is then important to accurately describe traditional methods such that the proposed machine learning based framework can be fairly compared to the current state-of-the-art.

2.1 Traditional Nuclear Material Accounting

Material Unaccounted For (MUF) [4] is a core component of NMA. MUF is a quantitative balance between flows of material into and out of a facility. Usually, facilities will have multiple material balances that are divided up to reach certain timeliness goals or due to physical constraints within a facility (e.g., separate buildings). MUF is calculated at regular intervals defined by the material balance period (MBP). Subject matter expertise is used to determine both the number and size of material balances in addition to the material balance period. The MUF calculation at a given time t with measurement locations i and total number of locations for a given measurement n is given by Equation 1.

$$MUF_T = \left(\sum_{i=1}^{n_I} I_{i,t-1} + \sum_{i=1}^{n_{inp}} Tin_{i,t} - \sum_{i=1}^{n_{out}} Tout_{i,t} \right) - \sum_{i=1}^{n_I} I_{i,t} \quad (1)$$

The individual terms in the equation are as follows:

- $\sum_{i=1}^{n_{inp}} Tin_{i,t}$: Total input transfers
- Transfer terms are often streams of material which should then be time integrated. The total transfer term would then become $\sum_{T=t-1}^{t} \sum_{i=1}^{n_{inp}} Tin_{i,t}$
- $\sum_{i=1}^{n_{out}} Tout_{i,t}$: Total output transfers
- $\sum_{i=1}^{n_I} I_{i,t}$: Total of all inventories at time t
- $\sum_{i=1}^{n_I} I_{i,t-1}$: Total of all inventories at time *t-1*

The expectation is that $MUF_t = 0$ when no material has been removed as all material has been accounted for. However, measurements always have some associated error, which causes a non-zero MUF even during normal conditions.

2.2 Measurement Error

No measurement is perfect and therefore is accompanied by some degree of uncertainty. Safeguards measurements are often characterized by a multiplicative error model as described in Equation 2.

$$M_{i,t} = G_{i,t}\left(1 + S_i + R_{i,t}\right) \quad (2)$$

Where

$$S_i \sim N(0, \delta_S^2)$$

$$R_{i,t} \sim N(0, \delta_R^2)$$

The terms above are defined as follows:

- $M_{i,t}$: Measured quantity of interest at location i at time t
- $G_{i,t}$: True quantity of interest (unobservable) at location i at time t
- S_i : Short-term systematic (i.e. epistemic) error
- Arises from measurement conditions or settings that remain unchanged from some period of time
- Difficult to reduce
- Example: Error due to calibration curve
- $R_{i,t}$: Random error (i.e. aleatory)
- Varies unpredictable under repeated conditions
- Can be reduced through repeated measurements
- Example: Counting statistics
- δ : Relative standard deviation

The random and systematic errors are assumed to be independent Gaussian random variables with zero mean and variances δ_S^2 and δ_R^2. Measurement errors are approximately normally distributed according to Equation 3. The specific values of the variances depend on the measurement technology that is used. The IAEA has published the International Target Value (ITV) guidelines [5] which provides expected performance metrics and variances.

$$M_{i,t} \sim N\left(G_{i,t}, G_{i,t}^2(\delta_R^2 + \delta_S^2)\right) \quad (3)$$

Measurement error plays an important role in the performance of anomaly detection for material losses. Generally, a material loss can be thought of as a mean shift in the normally distributed material balance, as expressed in Equation 4. A key goal of NMA is to detect this shift.

$$N\left(\mu_t \rightarrow \mu_t\ , \sigma_{MB}^2\right) \quad (4)$$

The body of statistics literature contains a range of different tests that can be used for change detection such as the one shown in Equation 4. However, all approaches are generally subject to limitations arising from measurement error as expressed in Equation 5. The probability of detection of a mean shift in a known, normal distribution (i.e. true positive) approaches the probability of false alarm (i.e. false positive) as the variance increases.

$$\lim_{\sigma_{MB} \to \infty} P\left(Alarm \vee N\left(\mu_t\ , \sigma_{MB}^2\right)\right) = P\left(Alarm \vee N(\mu_t, \sigma_{MB}^2)\right) \quad (5)$$

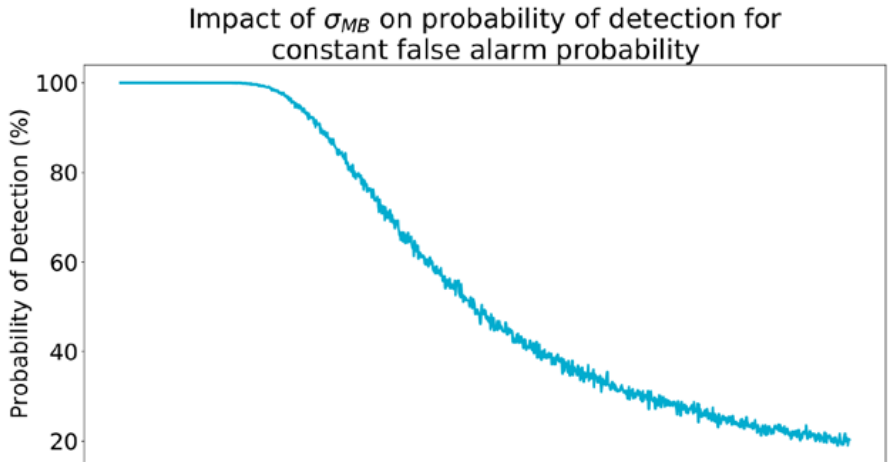


Figure 1: Probability of detection shown as a function of uncertainty for a constant false alarm probability.

Put simply, smaller mean shifts relative to the variance are more difficult to detect as they often get lost in the noise. This is shown more concretely in Figure 1 where the detection probability for an arbitrary, fixed material loss is quantified for a fixed false alarm probability and various levels of measurement uncertainty.

Finding strategies for reducing the material balance uncertainty has been a historical target for safeguards R&D given the impact on detection of material loss. One possible improvement would be to reduce the measured quantity size which would require a more frequent material balance period. This requires some optimization as too frequent material balance closures will result in higher false alarm probabilities [6]. Improving measurement uncertainty, which also reduces material balance uncertainty via smaller δ_S^2 and δ_R^2, is currently what drives the use of expensive DA measurements.

2.3 Sequential Material Balance Testing

Discussion so far has focused on a single material balance at a specific point in time. However, timely detection of potential material losses, a key goal of the IAEA, often requires multiple sequential material balances. For example, consider the case of a single yearly material balance where a diversion is initiated near the beginning of the year. Consequently, it would be months before the loss could be detected. Sequential material balances also have the added benefit of reducing the uncertainty of any single balance while noting that there are some restrictions on frequency of material balance closure.

Each individual material balance is comprised of potentially many normally distributed measurements which imply the material balance will also be normally distributed. As each single material balance has some mean and variance, a sequence of MBs can be expressed as a multivariate normal in Equation 6.

$$\overrightarrow{MB_t} \sim MVN(\overrightarrow{\mu_t}, \Sigma_t) \quad (6)$$

where

$$\overrightarrow{\mu_t} = (\mu_1, \mu_2, \ldots, \mu_t)^T$$

Recall that each individual material balance at time t is a function of the total current and previous inventories ($\sum_{i=1}^{n_I} I_{i,t}$ and $\sum_{i=1}^{n_I} I_{i,t-1}$). This results in a temporally correlated material balance sequence. However, the Standardized Independent Material Unaccounted For (SITMUF) transformation [7] can be used to decorrelate the sequence by accounting for the analytically determined covariance (using propagation of variance), Σ, and the conditional expectation with a few assumptions. Although not covered extensively here, traditional NMA relies on the residual between the observed MUF value and the conditional expectation of MUF. A sequential test, namely Page's trend test [8] [9], is used to detect trends in the material balance sequence residual. Under normal conditions, the SITMUF sequence should be approximately zero owing to a good conditional expectation. Material losses lead to poor expectations and larger residuals.

2.4 Machine Learning

Machine Learning (ML) refers to algorithms that perform a task without being explicitly programmed to do so. ML has seen a large surge in interest and is now embedded into many aspects of our daily lives. Although arguably less popular than domains such as computer vision, anomaly detection has benefited greatly from improvements in ML. Given the limitations described in previous sections, namely the dependence of traditional NMA on measurement uncertainty, it would be desirable to develop a ML framework that could sidestep the limitation. Specifically, a notable

improvement would be the use of lower cost, but higher uncertainty process monitoring (PM) and non-destructive assay (NDA) measurements to detect material loss. Such a framework would require framing material loss as an anomaly detection problem. This contrasts with traditional NMA which attempts to detect diversions through direct quantification of nuclear material (i.e. MUF).

There are many different anomaly detection algorithms that have been proposed as there is no universal solution for all problems. Consequently, this work represents only one potential, but informed solution for applied ML to improve nuclear material accountancy. Specifically, this work considers supervised regression with an unsupervised anomaly detection problem. The supervised regression problem requires the ground truth to learn an approximate function for some task. In this case, the regression task is to learn the behavior of parts of the PUREX reprocessing facility. Then, an unsupervised anomaly detection algorithm is used to detect unusual behavior. This class of anomaly detection algorithm does not require specific labelled examples of anomalies and instead relies on some proxy metric to describe normality. Unsupervised methods are particularly desirable for safeguards applications where it can be difficult or impossible to provide examples of all credible material loss pathways. This also facilitates a more direct comparison with the existing benchmark (Page's trend test on SITMUF) which also has no requirement with regards to examples of material loss.

In contrast, supervised approaches do require explicit, labelled examples of material loss, but do offer some potential advantages. For example, supervised approaches enable for direct optimization of material loss detection rather than specification of a proxy problem. Direct optimization through specific examples of material loss could also lead to better feature representation in supervised approaches leading to improved performance for known, high consequence loss pathways. Supervised approaches may prove useful, but were not considered in this work.

2.4.1 Related Work

Several previous works have attempted to develop improved strategies for guarding against material losses by developing novel approaches. One example is the Multi-Isotope Process Monitor (MIP) [10] wherein existing process monitoring measurements were combined with pattern recognition techniques in an attempt to develop more effective detection of material loss at large throughput facilities. MIP used principal component analysis (PCA) [11] to reduce the dimensionality of gamma a spectra to learn new representations that express most of the signal variance. Then, PCA statistics such as Q-residual could be used to detect anomalies. The approach used by MIP was limited by the linear reduction in dimensionality. Other works [12] [13] have sought to improve on commonly used trend tests used on SITMUF. For example, there have been prior attempts to use autoregressive moving average (ARMA) [14] models with SITMUF in an effort to detect material loss [12]. A companion work has also considered the application of supervised deep learning anomaly detection to detect material loss. That proposed work leverages a few examples of material loss in attempt to generally improve anomaly detection [15].

3. Problem Statement

Traditional statistics for nuclear material accountancy have a strong reliance on low measurement uncertainty to produce good probabilities of detection as shown previously in Figure 1. ML methods excel at finding subtle changes in signals that could indicate anomalous behavior. Ideally, a ML-based framework could utilize higher uncertainty (and potentially unattended) measurements to detect material loss at the same performance level as traditional approaches.

3.1 Process Modelling

Obtaining data from actual nuclear facilities is often impractical due to cost and limited availability. The Separation and Safeguards Performance Model (SSPM) [16] [17] PUREX flowsheet has been used to provide synthetic training, test, and validation data for the techniques described in this work. The model was developed for systems-level analysis of safeguards design for various facilities including UREX+, PUREX, gaseous enrichment, fuel fabrication, electrochemical reprocessing, and more. The model uses MATLAB Simulink to track elemental and isotopic material flows through various unit operations. Measurement blocks are used to simulate different types of measurements such as PM, NDA, and DA. Several common statistical tests used by the IAEA are also integrated into the model.

A PUREX SSPM flowsheet, based on a generic facility [18], is shown in Figure 2. The grey blocks represent the processing vessels throughout the plant and contain significant detail about inventories, timing of operations, filling/emptying sequences, etc. Signals connecting the blocks contain mass flow information for all nuclear material and bulk flows. The blue blocks represent measurement points which feed the traditional material balance calculation. The shaded regions (red, blue, and green) correspond to various prediction regions where neural networks are used to learn the area's behavior.

3.2 Baseline Machine Learning Approach

This work is motivated by the universal approximation theorem [19] which states that an arbitrary-width single layer neural network can approximate any well-behaved function. It is important to note that the theorem does not comment on the learnability of such well-behaved function.

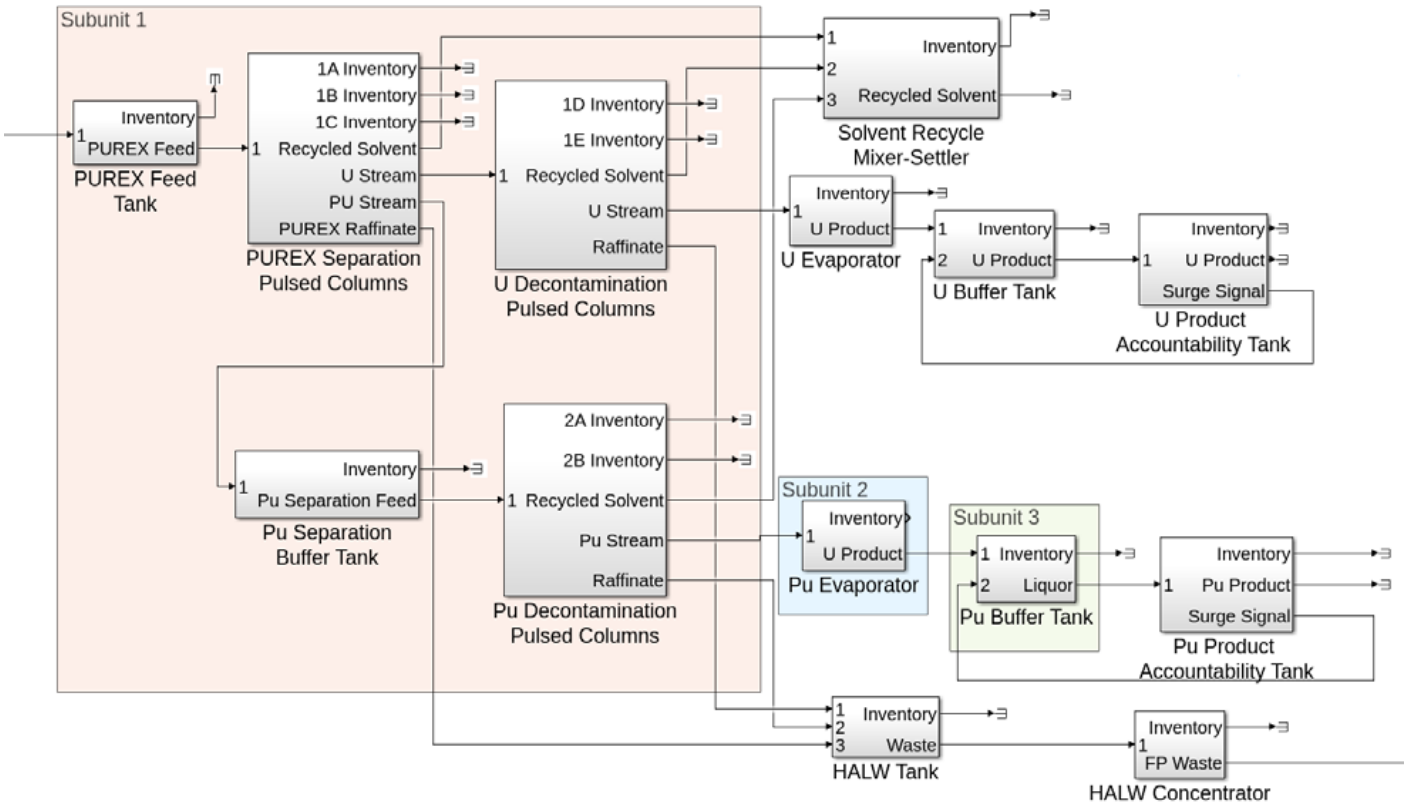


Figure 2: SSPM PUREX Model. Several labelled and shaded regions represent different areas of MBA2 that were learned by individual neural networks (i.e. subunits).

Nonetheless the hypothesis of this work is that a neural network should be able to learn the behavior of a large throughput nuclear facility, specifically a PUREX reprocessing facility. A material loss should appreciably change facility behavior such that the neural network will no longer provide accurate predictions. In turn, this will lead to discrepancies between observations and predictions that could be used to detect and possibly locate anomalous behavior (i.e. material loss). The hypothesis is summarized below in Figure 3.

The proposed unsupervised ML approach requires two steps. The first step is the prediction step where the neural network learns the behavior of a certain facility process (or area of processes) under normal conditions. Ideally, the neural network should be able to learn this behavior by way of the universal approximation theorem. Then, as the facility changes under anomalous conditions, the neural network predictions should no longer agree with observations. Figure 4 shows an example of this behavior wherein poor predictions are made during a window of anomalous behavior.

PUREX facilities have some operations that are time dependent and are not well suited to traditional feed forward neural networks, which have no temporal capacity. Consequently, the prediction step utilizes Long short-term memory (LSTM) [20] networks to complement traditional neural networks to capture the temporal properties of certain signals. For example, PUREX facilities have several mixing tanks that are dependent on material that has entered previously. Specific neural network architectures and data representation have a strong impact on accurate predictions. This work found that the LSTM networks trained well and produced good predictions when temporal behavior is captured by passing a window of history as input.

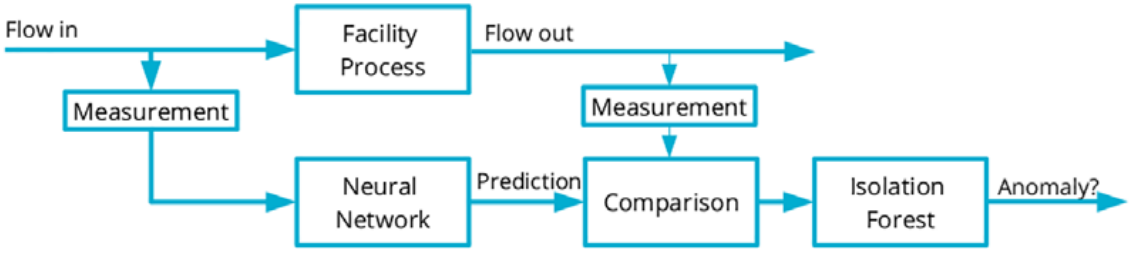


Figure 3: Proposed setup for applied ML for NMA

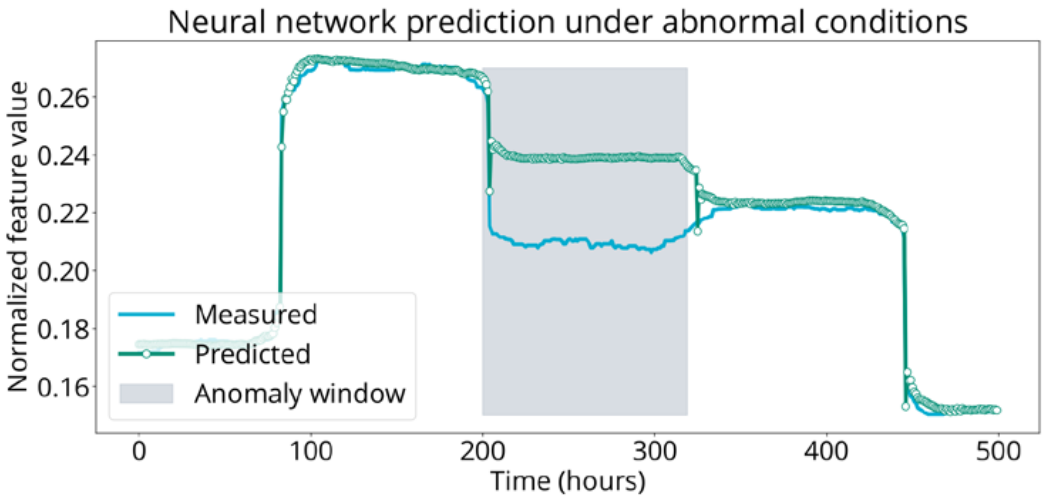


Figure 4: Neural network prediction during abnormal conditions.

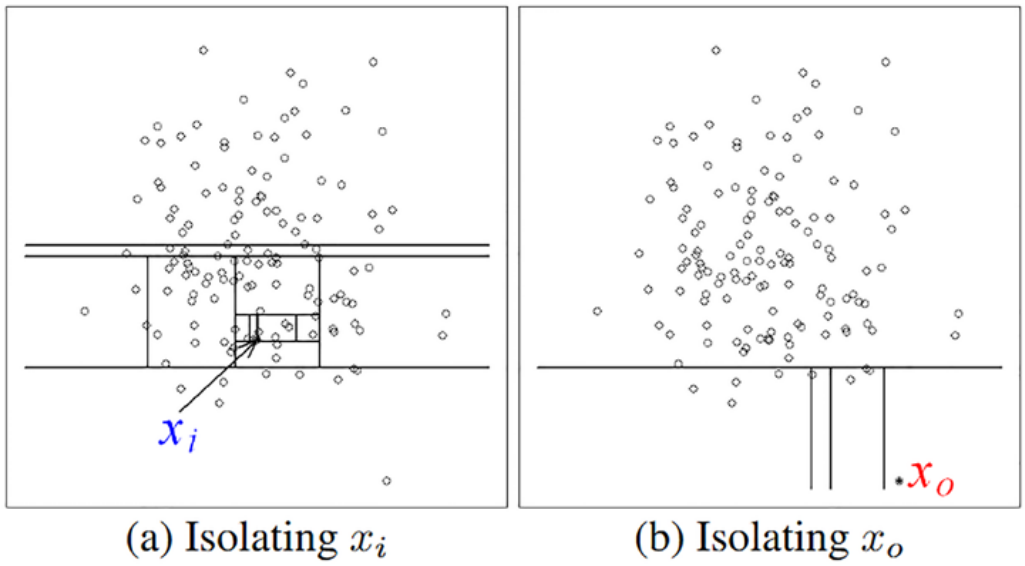


Figure 5: Isolation forest uses recursive splitting to measure the abnormality of a point. This figure shows a normal point, xi, which takes many splits to isolate it from the larger population. In contrast, the abnormal point x_0 requires fewer splits. [21].

The difference between the prediction and observed value, which in this work will be referred to as reconstruction error, is arbitrary due to imperfect predictions even under normal conditions. For example, the neural network used in the prediction step can never calculate predictions with full accuracy which always results in some non-zero prediction error. A second step is required to translate these arbitrary reconstruction errors into alarms and probabilities of detection. Identification of anomalous behavior is complex as PUREX facilities have large multidimensional datasets that arise from measurements at multiple locations each with several features. Instead of using a simple static threshold to detect anomalous behavior (e.g. alarm if a reconstruction error is greater than some scalar value), isolation forest is used here in combination with a classification window to define an alarm condition.

Isolation forest [21] is an unsupervised (requiring no examples of abnormal behavior) anomaly detection algorithm. The key intuition behind isolation forest is that anomalies should be few and different from normal data. The algorithm proceeds by selecting an observation, randomly selecting a feature, then randomly selecting a split value between the minimum and maximum. This process occurs recursively until the observation has been isolated from the larger dataset. Put simply, isolation forest will generate a list of logical criteria that make a particular observation appear unique. The criteria (i.e. splittings) can be represented as a tree structure. Gathering multiple sets of criteria results in a

forest, hence the name isolation forest. The path length of an observation averaged over several random trees is used as a proxy for normality. Points with path lengths below a threshold (as abnormal points should take less logical criteria to isolate) are considered anomalous. A visual intuition for isolation forest is shown in Figure 5.

Isolation forest has several hyperparameters that can be optimized through a grid search. These include the number of trees, maximum number of samples to train each estimator, and maximum number of features to train each estimator, which for this work, are set to 100, 15000, and 5 respectively. An additional hyperparameter, namely, the rate of contamination in the training dataset (i.e. percent of data estimated to be anomalous), cannot easily be discovered through a grid search.

This work generally assumes a 2% contamination rate even though the entire dataset is normal. Effectively, this forces classification of 2% of the training dataset as anomalous. The normal points that are classified as anomalous represent observations with the highest applied errors (i.e. errors drawn from the distribution tails). As classifications alone are insufficient for detecting anomalous behavior (as some normal points are classified as off normal), an alarm criterion on the classification is required. Using prior knowledge that material loss should be rare, it can be assumed that isolation forest will only infrequently produce false positives (i.e. points that are classified as abnormal but are normal). An example of isolation forest output for different anomalies is shown in Figure 6.

Note that there are some classifications being made as normal in the protracted anomaly shown in Figure 6. This is a function of a particular set of measurement realizations. As anomalies become more protracted and closer to the uncertainty bounds, the off-normal classifications become increasingly sparse. Eventually, the algorithm will no longer classify anomalies as off normal as the anomaly magnitude decreases below the uncertainty bounds. This creates the need for a classification window. Off-normal classifications that are dense should represent an anomaly; therefore, a certain number of off-normal classifications in a particular window should cause an alarm. For example, if 10 out of the last 15 classifications are off-normal then the alarm condition has been reached.

It should be noted that there is some dependency between the classification window and contamination rate. Although this work used a 2% contamination rate, there are a range of possible values (1-6%) that still resulted in good detection levels. However, it is important to adjust both parameters (contamination rate and classification window) in parallel. Often, a higher contamination rate still resulted in the same detection probabilities, but higher false alarm rates. Consequently, the classification window requires adjustment in conjunction with the contamination rate.

The proposed unsupervised machine learning approach can be summarized as follows:

- Stage 1: Neural networks are used to predict behavior of several locations within PUREX facility
- Stage 2a: Isolation forest uses reconstruction errors (i.e. prediction - observation) from all subunits as input to classify behavior as normal or off-normal
- Stage 2b: A threshold is applied over recent outputs from stage 2 (i.e. isolation forest). If there are many off-normal classifications recently then an alarm condition is reached

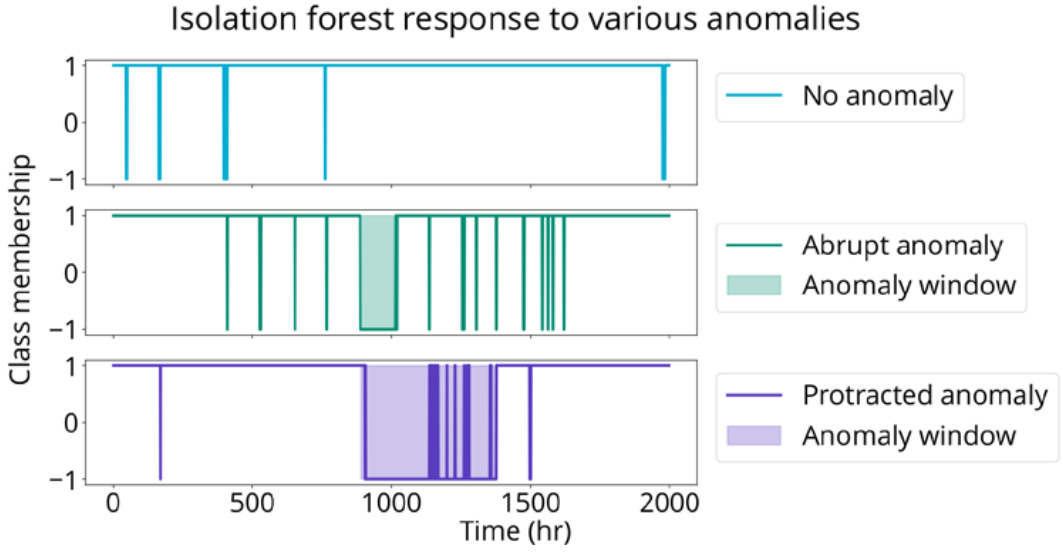


Figure 6: Isolation forest response to different anomalies. Class 1 is “Normal”, and class -1 is “Off-normal”.

4. Experimental Setup

The first step of this anomaly detection framework is to generate several datasets from the SSPM PUREX model to train and test the two-stage machine learning pipeline. The SSPM runs simulated randomized input fuel entering the facility to reflect real-world operation resulting in additional material flow and inventory variation. In practice, actual facilities will have a distribution of possible inputs and outputs rather than a single fixed input and output which results in a more difficult anomaly detection problem.

Each dataset generated by the SSPM model contains about 100 different runs to obtain good performance statistics for traditional approaches to benchmark against and to ensure sufficient data is available for training the machine learning algorithms. The runs are 6480 hours long (270 days), which is about one operating year for a PUREX facility. Ideally, this machine learning approach will operate directly on signals of interest (e.g. gamma spectra), however, the computational overhead of calculating tens of thousands of gamma spectra is large. Instead, this work considered mass with applied errors noting that this is not a direct evaluation of algorithmic performance. However, the use of mass to assess performance is a reasonable proxy as mass and gamma spectroscopy are related by a constant. The dataset evaluated in this work contained features representing ^{134}Cs, ^{137}Cs, ^{154}Eu, ^{241}Am, and ^{241}Pu in most cases. These features represent quantities that could realistically be observed at a PUREX facility.

Individual datasets for each location have a shape of [100 x 6480 x 5] where 100 is the number of runs (operational facility years), 6480 is the time in hours (assumed to be 270 operational days per year), and 5 is the number of features contained. The machine learning pipeline required several datasets to perform training and performance evaluations which is detailed below.

1. First stage (neural network prediction) is trained with a 0.75/0.25 split for training and validation
2. First stage generated training dataset of normal residuals
3. Second stage (isolation forest) is trained
4. Final datasets reflecting different scenarios are used to evaluate performance

The final step in the pipeline is evaluation of a normal dataset to determine a false alarm probability and several anomalous datasets to quantify detection performance. Specifically, four different anomalous scenarios of increasing difficulty are considered. Although not fully described here note that scenario 1 is the easiest to detect while scenario 4 is the most difficult. All datasets used in this evaluation had errors applied according to the multiplicative error model described in Section 2.2 to represent real world conditions more accurately.

5. Identified Performance Factors

Real world application introduces several challenges that impact model performance. It is important to identify and resolve these issues given the high consequence environment of safeguards. The impact of several specific factors and traditional machine learning requirements are explored in the following.

5.1 Facility Discretization

Early work centered on training a single neural network that could learn the facility behavior at all measured locations. However, several unique facility operations prevented effective training of this large neural network. This resulted in the adoption of a discretized approach that segmented the facility into smaller regional neural networks. The following sections describe the primary facility regions (referred to as "subunits") that resulted from the facility segmentation. This discretization of the facility had no observable performance impacts on the supervised regression task. That is, information contained in each subunit was sufficient for the task of predicting the feature value at the next time step.

5.1.1 Subunit 1: Pulsed Separation Columns

The first segment of the facility encompasses a region from the head end, where dissolved nuclear fuel enters the process, to the output of the decontamination column, where a purified plutonium solution leaves. Figure 2 shows this area highlighted in red.

This area consists of several continuous processing operations that are straightforward for a neural network to learn. Specifically, a bi-layer LSTM with a running history of input material is used to predict the output of the decontamination columns. The length of the input history was selected to be 200-hours, which was based on empirical performance as measured by mean-squared error (MSE) on the next time step prediction. The running history approach is expressed in Equation 7 where $f(x_t^n)$ is approximated by the bi-layer LSTM. Equation 7 denotes features as n and time as t. Use of LSTM layers is key to capture temporal dependencies between the inputs and outputs of this facility region.

$$
\begin{aligned}
&{}_1199 \leq t < \infty \\
&\overrightarrow{x_t^n} = [x_{t-199}^n, \ldots, x_t^n] \\
&\hat{x}_{t+1}^n = f\left(\overrightarrow{x_t^n}\right)
\end{aligned}
\tag{7}
$$

5.1.2 Subunit 2: Pu Evaporator

The second segment of the facility encompasses a single unit operation, namely the evaporator, or "Pu Evaporator" as shown in the blue region of Figure 2. This operation

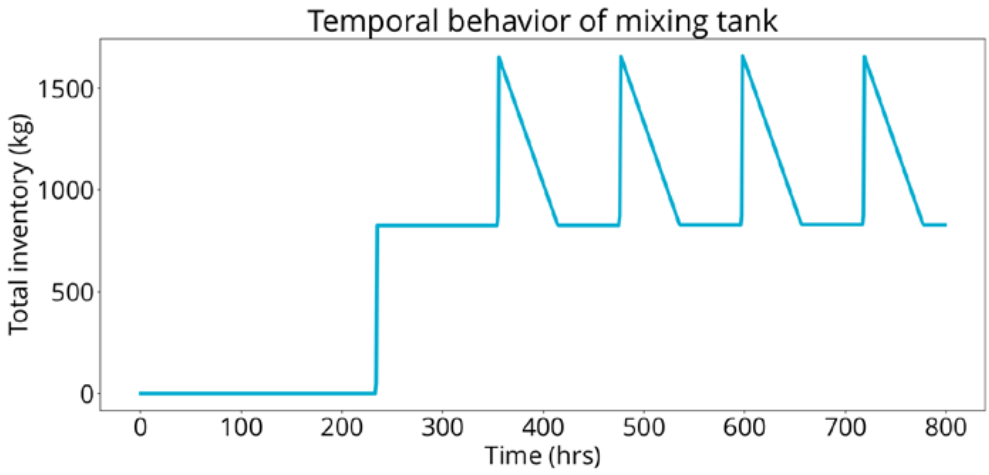


Figure 7: Mixing tank inventory

requires unique consideration as the signal is converted from continuous to discrete. During normal operation the evaporator accumulates solution until a setpoint is reached. Then, the evaporator reduces water content of the accumulated solution and outputs a discrete batch of material that is processed in a following operation.

The previous approach of a running history (described in Section 5.1.1) as input for a LSTM network is not appropriate for this area of the facility. For example, consider if the running history approach is used while the evaporator is accumulating solution and the setpoint had not been reached. The neural network would attempt to predict the previous output batch while the inventory reflects a different product. This area of the facility uses fixed window of time that precisely map the accumulated solution to the corresponding product. Here, a single layer feed-forward neural network is used to predict the output product given the total accumulated material

5.1.3 Subunit 3: Pu Buffer Tank

The final segment consists of another single unit operation, a mixing tank, listed as “Pu Buffer Tank” in the green region of Figure 2. This operation serves as a surge tank for the “Pu Accountability Tank”. During normal operation, this tank fills indefinitely, unless a surge signal is sent, in which case it empties. This behavior is regular and is shown in Figure 7. For this operation, the running history similar to what is described in Section 5.1.1 is used. A bi-layer LSTM network is again used; however, several hand-engineered features are required for the LSTM to make accurate predictions.

Note in Figure 7 that the tank output is a combination of two quantities: the previous batch of material to arrive in the tank and the residual tank inventory at the previous time step. Thus, a running average of the mixing tank inventory must be estimated. This running average can be roughly approximated as $(x_{t-1} + x_t)/2$. Additionally, the actual tank level measurement must also be included. During normal operations, there are slight variations in the input and output batch size, which will impact the running average calculation. Rather than adjusting the running average feature by hand, the LSTM can learn to adjust it during training provided that the tank level measurement, which is a function of the input and output sizes, is provided.

5.2 Training data availability

Machine learning algorithms often require large training datasets to demonstrate adequate performance at test time. One important factor driving required training data is model size. As the number of trainable weights and biases increases so does the training dataset size requirements. Safeguards data is often difficult to obtain, and real-world constraints could lead to little available training data. Therefore, it is important to consider how the dataset size impacts performance using concrete metrics. This section ignores measurement error (described in Section 5.4) to isolate the impact of available training data. As such, it is important to note these results would not reflect real-world results as there are other performance factors in addition to training data availability.

A parametric study is conducted to consider the impact of training dataset size on machine learning performance (probability of detection). It is important to note that both stages of the proposed approach (prediction and classification) require training data. Further, as the prediction stage is used to train the classification stage, there is a compounded effect of reduced training data. The classification stage will not only suffer from less training data, but poorer quality data as the prediction stage also degrades.

The baseline assumes 100 operational years of training data. This is chosen to be sufficiently large to ensure that training performance is driven by the machine learning algorithm hyperparameters which enabled fine tuning. It should be noted that this is not simply 100 iterations of the same operational year (i.e. same pattern with different errors applied), but unique simulations. There are a wide

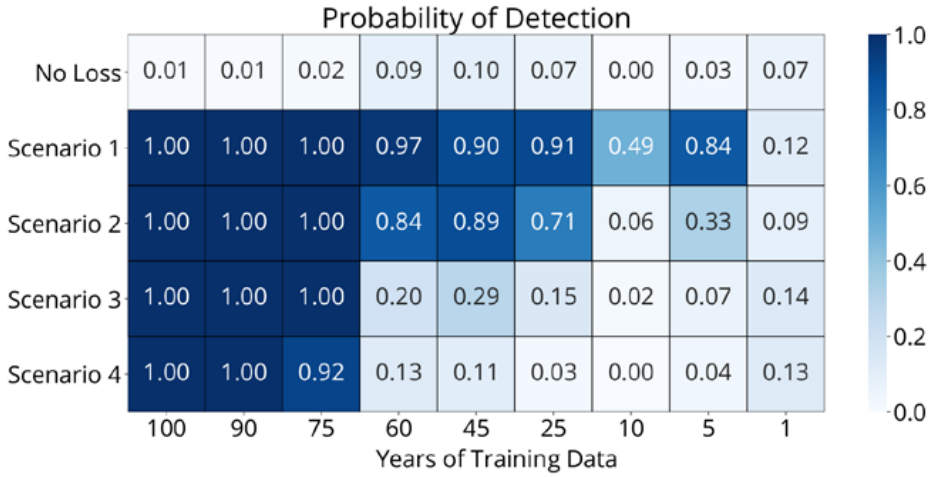


Figure 8: Probability of detection for several material loss scenarios with varied training dataset sizes. A probability of 1.00 indicates a 100% probability of detection whereas a probability of 0.00 indicates no probability of detection.

range of potential facility patterns due to the many combinations of input fuel that could be selected that are adequately captured in large datasets.

The parametric study considered the joint performance of reduced training data on the machine learning pipeline. That is, both stages are trained on reduced training datasets. For example, when 10 years of training data is used, the first prediction stage is trained on 10 years' worth of data. Then, 10 years' worth of predictions are generated and used to train the second stage. This essentially doubles the amount of training data that would be required in practice. This study also incorporated early stopping during training of the prediction stage to ensure that any performance losses are due to the inability of a smaller dataset to represent the test distribution rather than less training time. Results of this parametric study are shown in Figure 8.

Unsurprisingly, lower quantities of training data have a larger impact on the more difficult to detect scenarios. These scenarios tend to be relatively large changes compared to the uncertainty arising from measurement error. The more difficult scenarios have much lower performance while seeing sharp drop offs at certain quantities of training data. This is largely due to inflexible alarm threshold. Recall that the alarm criteria specified in this work is defined by a certain number of off-normal classifications within a window of time. Small changes in performance resulting in fewer off-normal classifications could result in large changes in alarm probabilities. For example, consider a threshold criterion of 30 off-normal classifications, sampled at a rate of once per hour, in a 50-hour window. A small degradation in performance that results in an average of 27 off-normal classifications when also sampled at the same rate in a 50-hour window translates to many fewer alarm triggers.

Poorer performance of the prediction stage also results in degraded detection performance for some of the subunit areas. The average prediction error for normal behavior generally increased with shrinking training dataset sizes (shown in Figure 9). A key assumption in this work is that a neural network can adequately learn facility operations. Increasing prediction errors from an inability to learn facility behavior results in a more difficult classification task.

It is interesting to note some subunits, which correspond to specific unit operations, are more susceptible to reduced training data than others. This phenomenon is not well understood and a target for future work. However, one possibility is that more complex operations require more training data, which is supported by machine learning literature. Often data requirements scale with both algorithm size and task complexity. Subunit 1 is a relatively simple area of the facility (pulsed separation columns) that degrades little with decreased training data whereas subunit 2 is more complex (evaporator) has a significant decrease in performance (i.e. higher errors).

5.3 Facility transients

Routine operations at bulk nuclear facilities can sometimes lead to transients that are not malicious in nature. These changes in behavior could make it difficult to detect anomalous behavior that occurs at the same time. However, a successful detection algorithm should be able to recover after the transient has ended and regain performance. Recovery of the proposed machine learning pipeline is evaluated by generating datasets representative of two different facility transients. The performance is reported as the reconstruction error (i.e. prediction-observation), which has strong correlations to probability of detection.

Facility transients can be grouped into several categories despite the numerous different potential scenarios. This work considers two different types of the transient. The first includes scenarios that change facility behavior but do not result in a new baseline. An example in this category might be small changes to product composition as a result of a

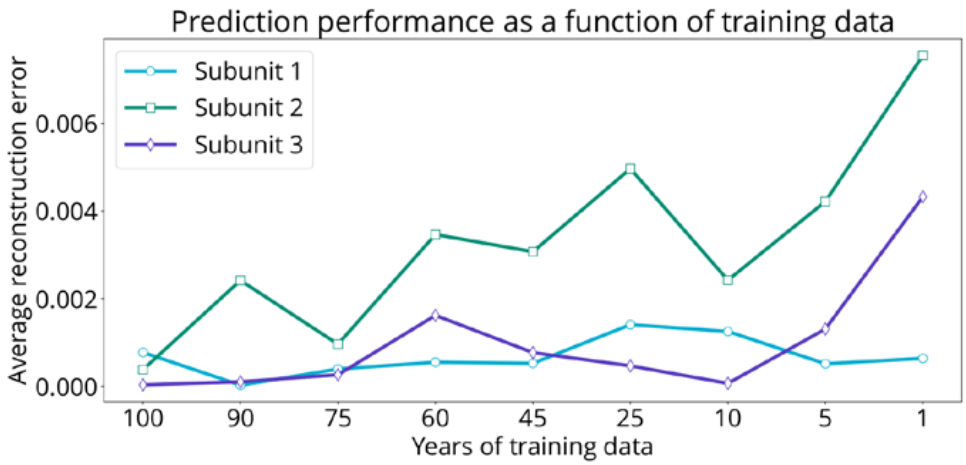


Figure 9: Impact of training data on prediction performance.

vessel leak. The second category includes scenarios that do significantly alter facility which results in a new baseline. Examples here would include transients that cause surge vessels to have a new equilibrium or changes to operational timing.

The first transient considered is in the first category where there is a temporary change in the behavior of the particular area. Figure 10 shows changes in the neural network prediction during and after this transient. The prediction is desirable during the transient in that the prediction error is high compared to normal behavior. However, after the transient has passed, the prediction no longer returns to a baseline performance level. This appears to indicate that the training data no longer represents conditions in the facility and that there has been some prolonged change in facility behavior. This is problematic as anomaly detection performance in the period of time after the transient will decrease.

The second transient represents a scenario where facility behavior is explicitly changed moving forward. For example, this could be damage to a unit operation or represent sensor failure. Figure 11 shows the prediction error during and after a simulated transient. Again, desirable behavior is shown during the transient where there is a sharp increase in prediction error. However, the prediction performance does not return to pre-transient levels. This is somewhat expected as the training data, which is represented to the algorithm as normal, no longer accurately represents the facility. Action would be required to adjust the prediction stage such that it reflects the updated normal conditions.

Both types of facility transients result in poor prolonged prediction performance, which would negatively impact probability of detection. While problematic, there exist some strategies within the machine learning literature to retrain algorithms in an online environment. Future work should target strategies to mitigate the impact of facility transients which are likely to occur in real-world scenarios.

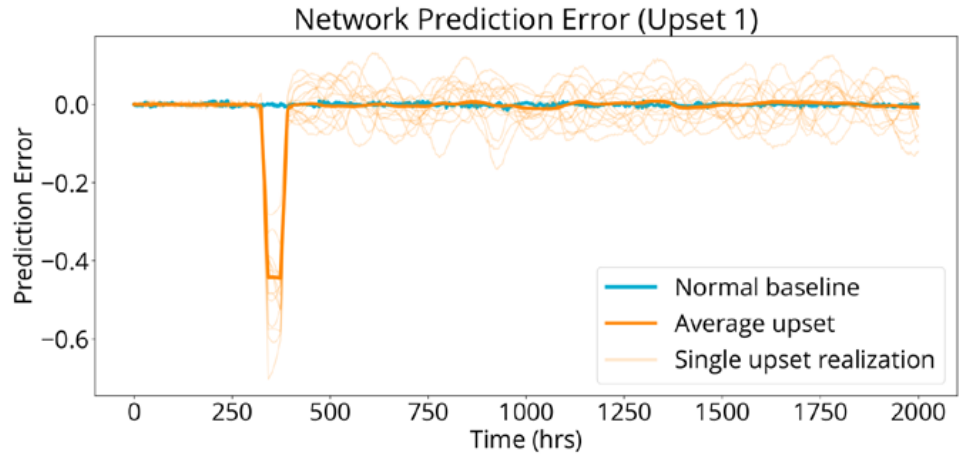


Figure 10: Prediction error of a neural network for a single isotope during a facility transient that changes the baseline of a facility section.

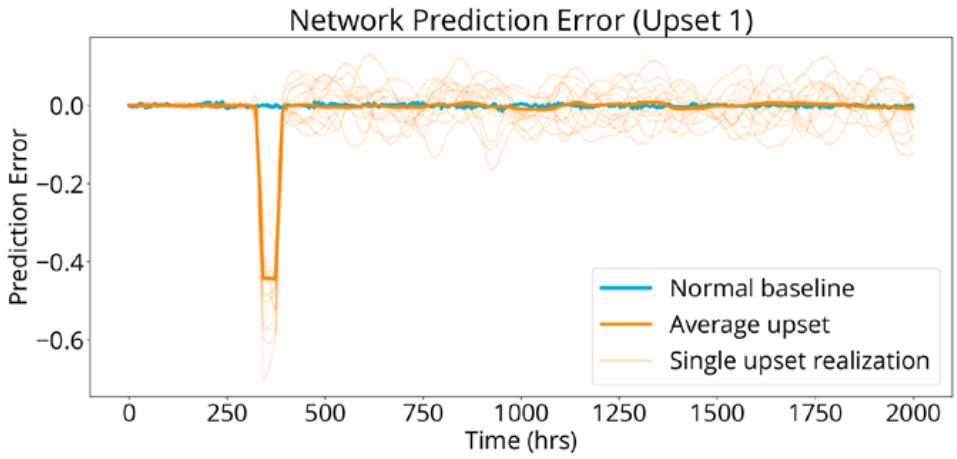


Figure 11: Prediction error of a neural network for a single isotope during a facility transient that changes facility behavior.

5.4 Measurement Error

Measurement error is a reality for the deployment of any real-world NMA system. Traditional NMA systems must implement specific strategies to detect losses when measurements are contaminated with error. For example, the SITMUF transformation can mitigate some impacts of measurement error by converting a MUF sequence to an uncorrelated sequence. Many common anomaly detection algorithms in the machine learning literature are prone to failure when used with error contaminated data. There are also few documented strategies on mitigating measurement error as most literature focuses on bias in supervised learning settings. For example, fairness which is an important area of research, seeks to remove human bias from collected datasets. However, this is fundamentally different from the multiplicative error model encountered in safeguards.

Fundamentally, detection of material loss (or any anomaly at all) is a mean shift detection problem. That is, given a normal distribution of features, can a shift in population mean be detected? This intuition forms the basis of several common anomaly detection algorithms. The proposed machine learning pipeline here also relies on a similar premise. Consider the training objective for the prediction stage; predict the facility behavior given some input. This is achieved through a mean squared error objective which attempts to minimize the difference between training examples and the prediction. It can be shown that the relationship in Equation 8 is true.

$$argmin_\theta \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i\theta)^2 \equiv argmin_\theta E_{x\sim \hat{p_{data}}}[logp_{model}(x)] \quad (8)$$

The mean squared objective is essentially the negative log-likelihood (i.e. cross-entropy) between the empirical distribution and a Gaussian model (i.e. the learned distribution, assumed to be normal). Effectively, the prediction stage tries to learn a function that produces an output distribution as close as possible to the training distribution. Then, under anomalous conditions, the learned distribution is no longer representative indicating that a mean shift has occurred.

Earlier, this work showed that increases in material balance uncertainty reduces the probability of detection for a material loss. A similar phenomenon is at play for the mean shift detection problem. Increases in a distribution's variance reduces the probability that a mean shift can be detected. This can be shown using a variety of approaches including a simple application of Bayes' theorem to a more complex analysis of variance (i.e. ANOVA) procedure.

The previous section showed that there is a strong dependence on sufficient large training datasets to achieve satisfactory anomaly detection performance. It is reasonable to assume, given the limited amount of safeguards data, that multiple measurement campaigns might be required to create a dataset of sufficient size. Each measurement campaign will have its' own unique set of calibrations (i.e. systematic error), that when aggregated together, will result in a larger variance than any individual dataset as shown in Figure 12.

The aggregation of multiple measurement campaigns results in the machine learning pipeline essentially learning variation due to measurement error in addition to facility behavior. This leads to lower anomaly detection performance than traditional statistical methods used for safeguards. This phenomenon is particularly nuanced and discussed at length in a companion work [22].

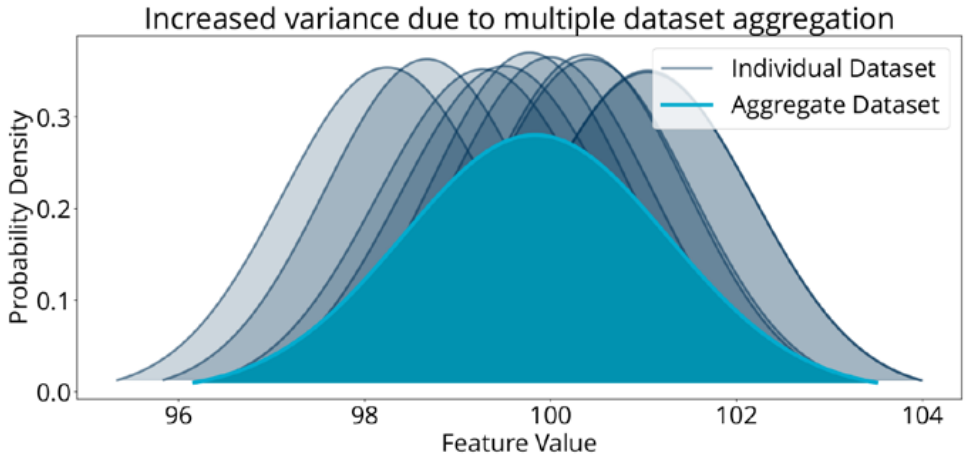


Figure 12: Probability density functions for multiple normal datasets.

5.5 Threshold selection

Recall that the second stage of this proposed machine learning approach requires some threshold for an alarm criterion. Isolation forest generates class labels when given the reconstruction error from the prediction stage. However, due to variation caused by measurement error, the classifications will never be perfect. Intuitively, material losses should generate more off-normal classifications (i.e. true positives) in each period as off-normal classifications made for normal observations (i.e. false positives) will be randomly distributed. One potential alarm condition would be requiring a specific number of off-normal classifications in a particular window of time. A common metric to tune safeguards thresholds (often defined by regulations to be 5%) is the false alarm probability (FAP, i.e. false positive rate).

Threshold optimization is underdefined in this case as there is one constraint (5% FAP) and two unknowns (window size and total classifications). This leads to multiple possible solutions for threshold criteria. In practice this has some impact on detection of abrupt material loss as shown in Figure 13. A parametric study is conducted that considered multiple combinations of windows and total classifications that resulted in a 5% FAP. The difference for most losses is insignificant but caused a 23% difference in detection probability for scenario 1, which is the most abrupt scenario. This reflects threshold where the total number of classifications is larger than the duration of the abrupt loss.

6. Ideal results

The previous section identified several factors that require attention to ensure adequate performance of the machine learning pipeline. Ideal (optimistic) performance can be quantified by accounting for these factors. The overall performance of the machine learning algorithm is compared to the traditional Page's trend test on SITMUF under near identical conditions in Figure 14. The ideal conditions used for Figure 14 made several assumptions:

- Sufficient training data available
- Optimal threshold selection
- No facility transients

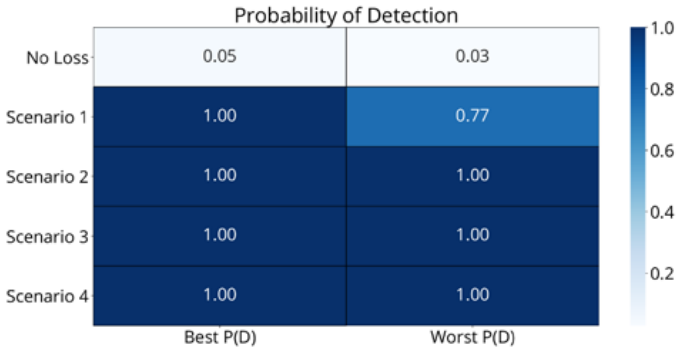


Figure 13: Probability of detection for several loss scenarios with varied thresholds.

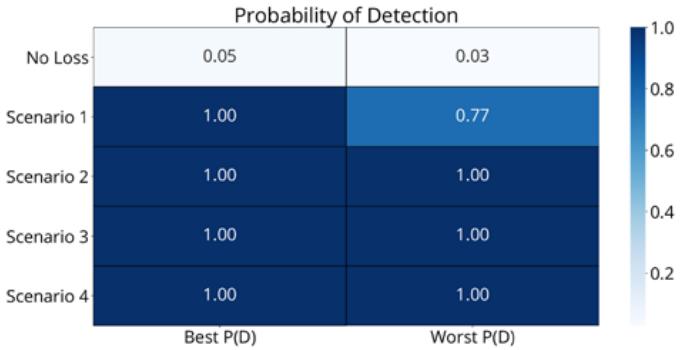


Figure 14: Detection probabilities for various loss scenarios.

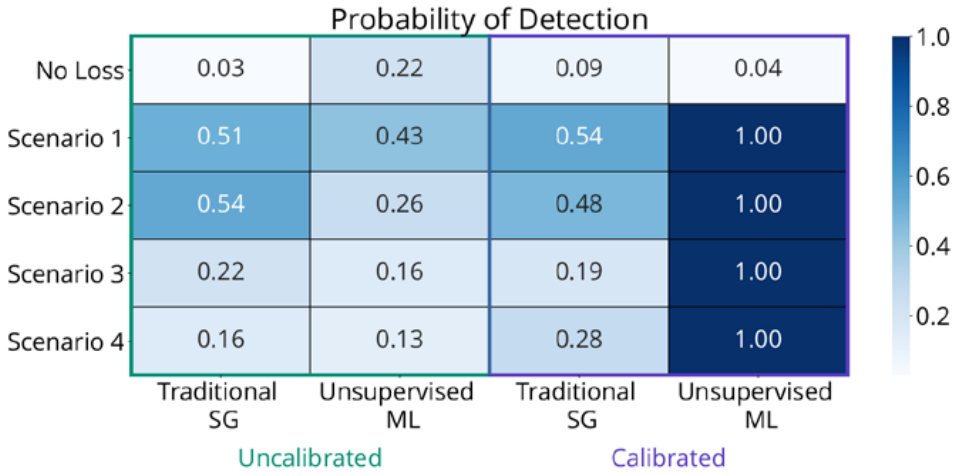


Figure 15: Detection probabilities for various loss scenarios.

Measurement error has many potential sources, and as such, could be difficult to resolve in real-world deployment scenarios. Consequently, it is not surprising that large portions of R&D for safeguards target reductions in measurement error. As there are no obvious data driven solutions to reduce the measurement error, Figure 15 considers the performance of both traditional statistical methods for safeguards and the unsupervised machine learning pipeline under "Uncalibrated" and "Calibrated" measurement conditions. Uncalibrated conditions are similar to current practices at facilities where sensors are placed and measured independently. The calibrated condition considers an experimental procedure wherein sensors are calibrated against each other (i.e. cross-calibrated). Here, the systematic errors for each sensor are the same non-zero value. For example, instead of having one sensor +1% biased and another being -2% biased, all sensors are biased at the same level.

The simulated calibration procedure has a large impact on the performance of the machine learning approach. Without it, performance is worse than traditional safeguards and very poor for most scenarios. It is interesting to note that the traditional safeguards approaches do not significantly benefit from this calibration procedure. This likely arises from implementation details for each approach. The machine learning algorithm is comparing signals from different locations in the facility, which is sensitive to mismatched biases specifically (i.e. large differences between sensors). However, the traditional safeguards approach is focused on quantifying MUF, which is sensitive to error in general.

7. Conclusions

This work proposed an unsupervised machine learning pipeline consisting of two steps to improve safeguards of bulk facilities. Several practical performance factors are crucial for real world performance. Several important findings are summarized below:

- Data representation is important to achieve adequate training performance
- Data availability has a significant impact on performance - Some facility operations are easier to learn than others and thus less susceptible to smaller training datasets
- Online training will likely be required after facility transients
- Measurement error has a significant negative impact on anomaly detection performance of the unsupervised machine learning approach
- The proposed alarm criteria are inflexible and can cause result in poor performance when not properly optimized. - A better criterion should be developed in future work.

Additionally, the generalization of the proposed pipeline was not studied in depth here. However, it is hypothesized that this approach will exhibit poor generalization even for facilities of the same type (I.e. other PUREX reprocessing facilities). The behavior learned through training will likely vary from facility to facility due to differences in equipment and facility layout. Applicability of common mitigation strategies for small datasets, such as transfer learning [23], to this problem remain unknown.

This work shows that unsupervised machine learning has the potential to out-perform traditional safeguards, but several requirements must be satisfied. There are several challenging limitations that are raised which make it unlikely that ML will wholly replace traditional safeguards in the near future. Data driven systems will likely complement existing safeguards systems until future work can resolve important barriers identified here.

8. Acknowledgements

This work was funded through the National Nuclear Security Administration's Office of International Nuclear Safeguards. The authors would also like to acknowledge the contributions of Michael R. Smith and Richard Fields (Sandia National Laboratories) who also contributed to this work. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2021-15805 J.

9. References

[1] United Nations, IAEA Statute, 1956.

[2] International Atomic Energy Agency, IAEA Safeguards Glossary, 2001 Edition, 2002.

[3] International Atomic Energy Agency, Establishing a System for Control of Nuclear Material for Nuclear Security Purposes at a Facility during Use, Storage and Movement, 2019.

[4] A. Goldman, R. Picard and J. Shipley, "Statistical Methods for Nuclear Material Safeguards: An Overview," Technometrics, vol. 24, pp. 267-275, 1982.

[5] K. Zhao, M. Penkin, C. Norman and S. Balsley, "International Target Values 2010 for Measurement Uncertainities in Safeguarding Nuclear Materials," ESARDA Bulletin, 2012.

[6] R. R. Picard, "Sequential analysis of material balances," Journal of Nuclear Materials Management, vol. 15, 1987.

[7] B. Jones, "Near real time material accountancy using SITMUF and a joint page's test: comparison with MUF and CUMUF tests," ESARDA Bulletin, vol. 15, pp. 20-26, 1988.

[8] E. S. Page, "Continuous Inspection Schemes," Biometrika, 1954.

[9] T. Burr and M. S. Hamada, "Revisiting Statistical Aspects of Nuclear Material Accounting," Science and Technology of Nuclear Installations, 2013.

[10] C. R. Orton, C. G. Fraga, R. N. Christensen and J. M. Schwantes, "Proof of concept experiments of the multi-isotope process monitor: An online, nondestructive, near real-time monitor for spent nuclear fuel reprocessing facilities," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 672, pp. 38-45, 2012.

[11] I. T. Jolliffe, "Principal Component Analysis and Factor Analysis," in Principal Component Analysis, 1986, pp. 115-128.

[12] T. Burr and M. S. Hamada, "Smoothing and Time Series Modeling of Nuclear Material Accounting Data for Protracted Diversion Detection," Nuclear Science and Engineering, pp. 307-320, 2017.

[13] S. Prasad, D. Booth, M. Y. Hu and S. Deligonul, "The Detection of Nuclear Material Losses," Decision Sciences, vol. 26, no. 2, pp. 265-281, 1995.

[14] E. J. Hannan, Multiple Time Series, Wiley, 1970.

[15] R. Gladen, T. Grimes, B. Wilson, D. Jack, N. Shoman and B. B. Cipiti, "Neural Assessment of Non-Destructive Assay for Material Accountancy," ESARDA Bulletin, vol. 63, 2021 (Preprint).

[16] B. Cipiti and N. Shoman, "Bulk Handling Facility Modeling and SImulation for Safeguards Analysis," Science and Technology of Nuclear Installations, 2018.

[17] B. B. Cipiti, "Process Monitoring Considerations for Reprocessing," in Institute of Nuclear Materials Management, 2015.

[18] R. J. Jones, W. R. Kane and M.-S. Lu, Detailed Description of an SSAC at the Facility Level for Irradiated Fuel Reprocessing Facilities, 1986.

[19] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," Mathematics of Control, Signals, and Systems, pp. 303-314, 1989.

[20] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," Neural Computation, 1997.

[21] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation Forest," in IEEE International Conference on Data Mining, 2008.

[22] N. Shoman and T. Burr, "Impact of Measurement Error on Deep Neural Networks for Nuclear Material," Nuclear Engineering and Design, 2021 (Preprint).

[23] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang and C. Liu, "A Survey on Deep Transfer Learning," in International Conference on Artificial Neural Networks, 2018.

NukeLM: Pre-Trained and Fine-Tuned Language Models for the Nuclear and Energy Domains

Lee Burke, Karl Pazdernik, Daniel Fortin, Benjamin Wilson and Rustam Goychayev
Pacific Northwest National Laboratory
902 Battelle Blvd, Richland, WA 99354
E-mail: lee.burke@pnnl.gov

John Mattingly
North Carolina State University
Raleigh, NC 27695

Abstract:

Natural language processing (NLP) tasks (text classification, named entity recognition, etc.) have seen revolutionary improvements over the last few years. This is due to language models such as BERT that achieve deep knowledge transfer by using a large pre-trained model, then fine-tuning the model on specific tasks. The BERT architecture has shown even better performance on domain-specific tasks when the model is pre-trained using domain-relevant texts. Inspired by these recent advancements, we have developed NukeLM, a nuclear-domain language model pre-trained on 1.5 million abstracts from the U.S. Department of Energy Office of Scientific and Technical Information (OSTI) database. This NukeLM model is then fine-tuned for the classification of research articles into either binary classes (related to the nuclear fuel cycle [NFC] or not) or multiple categories related to the subject of the article. We show that continued pre-training of a BERT-style architecture prior to fine-tuning yields greater performance on both article classification tasks. This information is critical for properly triaging manuscripts, a necessary task for better understanding citation networks that publish in the nuclear space, and for uncovering new areas of research in the nuclear (or nuclear-relevant) domains.

Keywords: nuclear; energy; language; classification

1. Introduction

While natural language processing (NLP) has made significant strides in recent years, its application to the nuclear domain has remained rudimentary. In any domain, the ability to classify and prioritize information is critical when the data volume is large and growing. To enable the discovery of new connections between existing technologies or the potential use of a new technology in the nuclear domain, simple keyword searches are insufficient. To accelerate research in the nuclear domain, a language model is needed—one that "understands" nuclear terminology, "understands" terminology in similar energy domains, and can automatically uncover latent similarities between materials, methodologies, and technologies.

In addition to accelerating nuclear science, this new methodology would be valuable to the International Atomic Energy Agency (IAEA) as part of their information collection and processing system. Quantifying the threat of a nation state's nuclear capability presents a particularly complex problem because the use, development, and transfer of nuclear technology is not itself an indication of nefarious intent. Technology itself has the added complexity of encompassing both physical items of trade, as well as social networks in academia and industry settings, where the "technology" is not a physical, tradeable good, but the knowledge and capabilities of individuals [1]. Further, as international scientific collaborations become more prevalent, transfer of nuclear technology may become more prevalent, including inadvertent transfers. Readily available open-source information about such research collaborations, e.g., journal papers and technical reports, can offer indications of the use or transfer of such technology. Extant approaches to processing such information, to the limited extent it is attempted, rely heavily on manual analysis by humans, a method constrained by time and subject-matter expertise. A new approach would help the IAEA to develop capabilities toward the detection of nuclear technology use or transfer through analysis of technical publications.

The amazing progress of state-of-the-art NLP methods has opened up new opportunities for nuclear domain

researchers to leverage powerful language models. Models like BERT [2] have shown significant improvement in NLP benchmarking metrics, such as the General Language Understanding Evaluation (GLUE) benchmark [3]. These benchmark metrics evaluate a language model's ability to perform a variety of tasks that resemble human ability to comprehend and be language literate. Though undoubtedly one element of BERT's success is its large architecture of stacked Transformers [4], another is the widespread use of transfer learning: pre-training on one task then fine-tuning on another. By pre-training on general-purpose corpora, a model has a strong foundation when approaching particular benchmark tasks.

There is also evidence that the performance of pre-trained language models on some tasks can be improved even further by domain-adaptive pre-training [5]—that is, starting with a model pre-trained on general-purpose corpora, then continuing the pre-training process on a corpus that is more representative of the domain of interest.

Given the recent success of large, Transformer-based neural network architectures and domain-adaptive pre-training, as well as the need for nuclear-"aware" NLP models, we have developed NukeLM, a language model trained on nuclear-relevant research that performs best on nuclear-relevant downstream tasks.

2. Related Work

A number of scientific and computational advances in recent years have led to significant improvements in the performance of computational models for natural language inference and understanding. Notable among these is the field of transfer learning, using pre-trained models for downstream tasks perhaps markedly different from their original tasks. Often, this takes the form of semi-supervised learning, where a model is trained on un-labeled data using a self-supervised task, then fine-tuned on a supervised task in the same domain.

Word embeddings (e.g., word2vec [6], GloVe [7], fastText [8]) learn a projection from the high-dimensional vocabulary space of a corpus of texts into a much smaller vector space using self-supervised training tasks like predicting nearby words. A key drawback of this approach is that each word is associated with a single vector, regardless of context.

A number of approaches have been proposed to learn contextualized word embeddings. For instance, ELMo [9] trains separate forward- and backward-oriented models for next-word prediction, then learns linear combinations of the deep representations for downstream tasks. In contrast, BERT [2] learns to encode context from both left and right at once using a very large architecture of stacked Transformers [4], pre-training with both a word prediction task (masked language modeling, MLM) and a task to predict whether a given sample follows another in the original text, relative to being chosen randomly from the corpus (next-sentence prediction, NSP).

RoBERTa [10] leverages the same Transformer-based architecture as BERT, but shows improvements on downstream tasks with some changes to its pre-training strategy: it removes the NSP objective, pre-training only with MLM; it allow samples to cross document boundaries in pre-training, ensuring all pre-training samples are as long as possible; it determines which tokens to predict in each batch rather than deciding offline, before training; it uses much larger batch sizes; it uses byte-level tokenization instead of character-level; and finally, it considers much more pre-training data, including those from the Common Crawl corpora.

SciBERT [11] clones BERT's stacked Transformer architecture and pre-training methodology but replaces the BERT training corpus with a large, multi-domain corpus of scientific publications. This results in better performance on scientific domain tasks because of the better match between the domains of pre-training and fine-tuning tasks.

In contrast to training a domain-specific model from scratch like SciBERT, Gururangan et al. [5] demonstrate that continued pre-training of a general-purpose language model on in-domain text (called domain-adaptive pre-training, DAPT) can lead to improved performance on downstream tasks, but that continued pre-training on out-of-domain text can worsen performance. They explore several ways to bootstrap a targeted continued-pre-training corpus and explore the tradeoff between performance and computational expense.

Similarly, several domain-specific models have been proposed that continue pre-training from a BERT checkpoint. BioBERT [12] continues pre-training on biomedical corpora. NukeBERT [13] continues pre-training on a nuclear-domain corpus, with the addition of newly initialized vocabulary entries specific to the nuclear domain. However, in contrast to NukeLM, the pre-training corpus for the NukeBERT model was generated from a relatively small corpus consisting of about 7000 internal reports from the Indira Gandhi Centre for Atomic Research, largely focused on fast breeder reactors; the NukeBERT language model is somewhat narrowly focused on nuclear reactor research for power generation rather than defining topics broadly associated with the nuclear fuel cycle. Furthermore, it is not clear if the NukeBERT language model is publicly available, and the associated dataset is not available under a standard open-source license.

3. Data

We consider scientific abstracts from the U. S. Department of Energy Office (DOE) Scientific and Technical Information (OSTI) database [14] obtained in November 2018, amounting to nearly two million abstracts from over 70 years of research results from DOE and its predecessor agencies. No pre-processing is performed on these abstracts; they are analyzed as they appear in the database.

For fine-tuning, we consider only abstracts labeled with a subject category. The possible categories are formalized by OSTI, and all products submitted to OSTI are encouraged to provide at least one, listing the primary category first. If more than one category is specified, we consider only the first. In addition to the multi-class labels induced by the OSTI subject categories, we formulate binary labels by identifying OSTI subject categories that correspond to the top level of the IAEA Physical Model [15], which describes acquisition pathways. The topics described in the IAEA Physical Model include ore mining and milling, pre-conversion, uranium enrichment, post-conversion, fuel fabrication, nuclear reactors, heavy water production, and reprocessing of irradiated fuels. Using this criterion, the following OSTI topic categories are considered related to the nuclear fuel cycle for the binary classifier: nuclear fuels, isotope and radiation sources, nuclear fuel cycle and fuel materials, management of radioactive and nonradioactive wastes from nuclear facilities, specific nuclear reactors and associated plants, general studies of nuclear reactors, radiation chemistry, instruments related to nuclear science and technology, and nuclear physics and radiation physics. These categories are all assigned to the positive class in the binary classification problem ("NFC-related", referring to the nuclear fuel cycle [NFC]), regardless of the step or steps of the Physical Model to which they correspond. The list of all OSTI categories and their binary categorization designation is provided in Appendix A.

4. Experimental Setup

We begin with pre-trained checkpoints implemented in HuggingFace's transformers framework [16], available from the HuggingFace model database with the following slugs: roberta-base and roberta-large are base and large versions of the RoBERTa model, respectively, and allenai/scibert_scivocab_uncased is the recommended uncased version (i.e., inputs are converted to lower case) of SciBERT.

Following Gururangan et al. [5], we perform domain-adaptive pre-training. We continue pre-training all three models, SciBERT, RoBERTa Base, and RoBERTa Large, on 80% of the OSTI abstracts. For the remainder of this manuscript, we use the naming convention NukeLM to define RoBERTa Large with continued pre-training on OSTI abstracts. The remaining 20% of documents are held out from the pre-training process and split evenly into two data sets (200 K each) to be used for fine-tuning and testing the classification models. When forming each batch, 512-token segments are taken irrespective of document boundaries, and 15% of the tokens are masked for prediction. We train for 13 K steps with a batch size of 256, for a total of 3.3 M samples consisting of 1.7 B tokens (similar in size to the corpora in Gururangan et al. [5]). Other hyperparameters follow Gururangan et al. [5].

We perform some exploratory analysis of the impact of domain-adaptive pre-training on OSTI abstracts, including performance metrics and an example of masked word modeling.

For fine-tuning, we begin with the six models described above: RoBERTa Base and Large and SciBERT, both with and without OSTI domain-adaptive pre-training. We then follow Gururangan et al. [5] by passing the final layer [CLS] token representation to a task-specific fully connected layer for prediction (see the transformers documentation for details). A validation set is held out, consisting of 10% of the overall fine-tuning set.

We consider two tasks: multi-class prediction over the original OSTI subject categories, and binary prediction over the relevance of an abstract's subject category to one of the steps of the nuclear fuel cycle. The fine-tuning data set consisted of 198,564 documents, of which 23,268 are related to the nuclear fuel cycle according to our definition.

A small hyperparameter search is performed on the binary task (details in Appendix B), selecting a learning rate of 10^{-5} and a batch size of 64. We train for five epochs (14.7 K steps), evaluating at 20 checkpoints (about every 750 steps) and saving the best model according to loss on the validation set. Other hyperparameters follow Gururangan et al. [5].

5. Results of the Language Modelling Task

5.1 Metrics

The MLM task is evaluated based on the categorical cross-entropy between the one-hot true distribution over a model's vocabulary and a model's predicted distribution. This MLM loss is shown before and after domain-adaptive pre-training for each of the three baseline models in Table 1. As in RoBERTa-style pre-training, one token per sample is masked randomly, without consideration of sub-word status, stop words, or other factors.

Continued pre-training improves the performance of RoBERTa Base more than that of SciBERT, to the point where it performs better than the much larger RoBERTa without continued pre-training. The RoBERTa pre-training strategies may have yielded an easier-to-train model than the SciBERT methodologies, but this may be due solely to the larger vocabulary size, 50 K tokens for RoBERTa vs. 30 K

for SciBERT. Regardless, NukeLM shows improvement over RoBERTa Large, and remains the most accurate of the models.

Model	MLM Loss
RoBERTa Base	1.39
RoBERTa Base + OSTI	1.11
RoBERTa Large	1.13
NukeLM	**0.95**
SciBERT	1.34
SciBERT + OSTI	1.18

Table 1: Masked language modeling loss, based on categorical cross-entropy between true and predicted probability distributions, on the evaluation sub-set of the OSTI pre-training data. Lower is better. The symbol "+ OSTI" denotes continued pre-training on OSTI abstracts. The best performing model is in **bold**.

Model	Top-5 Predictions	Score
RoBERTa Base	metal	0.252
	metals	0.149
	uranium	0.145
	water	**0.130**
	iron	0.026
RoBERTa Base + OSTI	**water**	**0.955**
	metal	0.008
	elements	0.008
	metals	0.008
	oil	0.003
RoBERTa Large	**water**	**0.951**
	metal	0.013
	metals	0.011
	fuel	0.004
	carbon	0.002
NukeLM	**water**	**0.996**
	metals	0.001
	oil	0.001
	#water	<0.001
	metal	<0.001
SciBERT	metal	0.225
	metals	0.117
	water	**0.068**
	iron	0.052
	argon	0.042
SciBERT + OSTI	**water**	**0.929**
	metal	0.024
	metals	0.011
	iron	0.003
	oil	0.003

Table 2: An example of masked language modeling. Column 2 contains the top five tokens considered most likely (the true token, "water", is in **bold**), and column 3 contains the associated likelihood scores (the highest confidence for the true token is also in **bold**). The character "#" indicates the token is a sub-word, i.e., a prediction of "heavywater" rather than "heavy water". The symbol "+ OSTI" denotes continued pre-training on OSTI abstracts.

5.2 MLM Example

We present an example of masked language modeling to illustrate the task and performance improvement after domain-adaptive pre-training. The bolded word is masked, and the models are asked to predict what word should fill in the blank.

> The use of heavy **water** as the moderator is the key to the PHWR system, enabling the use of natural uranium as the fuel (in the form of ceramic UO2), which means that it can be operated without expensive uranium enrichment facilities. [17]

Table 2 summarizes the top five predicted tokens and their associated likelihood score from each of the six models after domain-adaptive pre-training (if any) but before fine-tuning. Before continued pre-training, all three models include the correct answer in their top five predictions, but RoBERTa Base and SciBERT predict the more common but incorrect phrase "heavy metal," albeit with low confidence; only RoBERTa Large predicts the correct answer, evidence that its large size allowed it to learn from pre-training alone some subtleties of the nuclear domain that the smaller models did not. After continued pre-training, all three models regardless of size succeed in predicting the correct answer with high confidence.

6. Results of Downstream Tasks

6.1 Multi-Class Classification

The results of fine-tuning of the multi-class classification task are presented in Table 3. SciBERT's advantage over RoBERTa Base persists after domain-adaptive pre-training, perhaps because its scientific-domain pre-training corpora are more closely related to the OSTI task than are RoBERTa's. However, neither overcomes RoBERTa Large even without the added advantage of continued pre-training, likely because the latter contains several times more trainable parameters.

Model	Accuracy	Precision	Recall	F1-Score
RoBERTa Base	0.6745	0.6564	0.6745	0.6603
RoBERTa Base + OSTI	0.6972	0.6884	0.6972	0.6863
RoBERTa Large	0.7056	0.7008	0.7056	0.7013
NukeLM	**0.7201**	**0.7164**	**0.7201**	**0.7168**
SciBERT	0.6972	0.6866	0.6972	0.6883
SciBERT + OSTI	0.7047	0.6981	0.7047	0.6973

Table 3: Results of fine-tuning on the multi-class classification task. Precision, Recall, and F1-scores are an average of all classes,

weighted by class size. The best performing model by each metric is presented in **bold**.

6.2 Binary Classification

The results of fine-tuning on the binary classification task are presented in Table 4. Without domain-adaptive pre-training, SciBERT performs even better than RoBERTa Large, possibly because of its more closely related pre-training corpora. However, unlike in the multi-class task, both SciBERT and RoBERTa Base see degraded recall (and, in the case of SciBERT, accuracy), outweighed by a moderate increase in precision only due to class imbalance. Only NukeLM sees improvement across all measured metrics, likely due again to its large size. It is worth noting that the much smaller RoBERTa Base is able to achieve performance comparable to the unwieldy RoBERTa Large via continued pre-training, which may be useful in resource-constrained applications.

Model	Accuracy	Precision	Recall	F1-Score
RoBERTa Base	0.9506	0.7938	0.7816	0.7876
RoBERTa Base + OSTI	0.9544	0.8237	0.7773	0.7998
RoBERTa Large	0.9506	0.7995	0.7722	0.7856
NukeLM	**0.9573**	0.8270	**0.8038**	**0.8152**
SciBERT	0.9548	0.8061	0.7910	0.7984
SciBERT + OSTI	0.9532	**0.8285**	0.7747	0.8007

Table 4: Results of fine-tuning on the binary classification task. Precision, Recall, and F1-scores consider NFC-related to be the positive class. The best performing model by each metric is presented in **bold**.

Moreover, numerically small improvements in performance metrics belie the very large size of the datasets presented here. An analyst attempting to filter a corpus as large as OSTI into a more manageable size would be well-served to choose NukeLM over the other models discussed above; a single percentage point change could translate to thousands of relevant papers that would have been missed, or irrelevant papers requiring manual inspection. Indeed, this use-case motivates a preference for recall (the fraction of true positives predicted to be positive) over precision (the fraction of predicted positives which are truly positive), further widening NukeLM's advantage over its competitors in our quantitative assessments.

6.3 Performance under Different Training Set Sizes

One reported advantage of domain-adapted language models is the ability to fine-tune on smaller numbers of labeled examples. We test this ability with the binary classification task described above. We randomly select increasingly large proportions of the binary classification fine-tuning set, ignoring the rest, so that each larger subset contains the earlier, smaller subsets. We train the off-the-shelf RoBERTa Large and NukeLM with the same experimental set-up as in Section 6.2 and track the log-loss computed on the hold-out evaluation set. This metric is computed via the Kullback-Leibler divergence, a measure of dissimilarity between the true and predicted probability distributions over the output categories, averaged over the test set. Twenty repetitions with different random seeds are performed. For visual convenience, the probability density function of each of these sets of repetitions is estimated using the kernel density estimation technique, analogous to a smoothed histogram. Generally, lower log-loss indicates better predictions, and greater separation of distribution

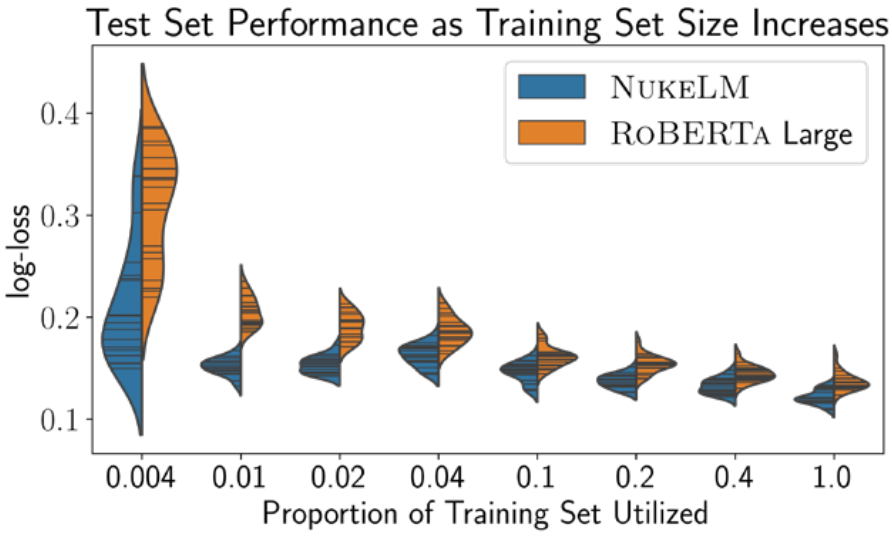


Figure 1: Binary classification performance, measured by log-loss on a hold-out test set, as the training set size is increased, for RoBERTa Large (orange) and NukeLM (blue). Hash marks are each of 20 repetitions with different random seeds, while filled areas are kernel density estimations.

density indicates more significant differences. Figure 1 summarizes the results.

While Table 4 summarizes some performance metrics of NukeLM compared to other models on the full fine-tuning data set for the binary classification task, this experiment provides insight into the potential benefit of using NukeLM on other fine-tuning tasks where the amount of labeled training data is likely to be on the order of one thousand and not one hundred thousand. The domain-adapted model achieves significantly better performance with smaller amounts of data; however, this advantage shrinks as the fine-tuning set size increases. This could be the result of continued pre-training priming the model for performance in this domain.

Therefore, the question of significance between NukeLM and other models is best understood as a function of fine-tuning training set size. Moreover, Figure 1 shows that even though the performance gain decreases with increasing amount of fine-tuning data, we still observe superior performance using NukeLM with a fine-tuning set of nearly two hundred thousand documents.

Interestingly, the disparity between models is less apparent at the lowest training set size tested (0.4% of the full corpus, or 754 documents). While NukeLM maintains its superiority, with so few examples used for fine-tuning, significant instability is observed over the repetitions. A follow-up experiment implements several strategies for stabilizing fine-tuning of large language models discussed in Zhang, et al. [18], but none have a major impact (see Appendix C for details) and are not employed further.

6.4 Qualitative Assessment

Beyond model performance on the MLM task and document classification, an important question regarding these trained language models is whether any reasonable interpretation can be made of the intermediate representations of input examples. While there is not a clear consensus on how useful these embeddings can be in providing explanations, with arguments from both sides [19, 20], there is undoubtedly some information contained within these transformer-based language models because their predictive ability is state-of-the-art. So, while a direct interpretation of an embedding produced by NukeLM may be questionable, the transformation of this high-dimensional space that results from pre-training should provide some explanation as to how prediction was improved.

As a first step toward interpreting the impact of domain-adaptive pre-training, we consider models fine-tuned on the binary classification task and visualize output embeddings from the most accurate models, RoBERTa Large both with (i.e., NukeLM) and without continued pre-training on OSTI abstracts. We use uniform manifold approximation and projection (UMAP) [21] with all default parameters to project the output corresponding to the special token [CLS] down to two dimensions, training separate UMAP projections for each model. Figure 2 (top row) depicts the result of this process performed on a 1000-sample random subset of the binary classification task validation set.

In both models, the positive class is generally clustered together; indeed, both models are able to learn relatively accurate decision boundaries. However, in the version without domain-adaptive pre-training, the cluster looks like a single manifold, eventually connecting to the mass of negative samples like an isthmus. In contrast, continued pre-training appears to encourage the model to form more complicated structures, with an isolated cluster of mostly positive samples in addition to a similar but much smaller isthmus connected to a large mass of negative samples.

To explore these differences further, we apply BERTopic [22], a clustering and topic modeling approach for understanding the output embeddings of a transformer model. BERTopic also uses a UMAP projection for dimension reduction, in this case to 100 dimensions, but then uses hierarchical density-based spatial clustering of applications with noise (HDBSCAN) [23] to cluster documents and a class-based TF-IDF (cb-TF-IDF) score for topic modeling. TF-IDF stands for term frequency and inverse document frequency, a standard method for identifying terms used unusually frequently in each document. Here, all documents within the same cluster are concatenated into a single document and then the usual TF-IDF score [24] is computed as follows:

$$\mathrm{cb-TF-IDF}_i = \frac{t_i}{w_i} \times \log \frac{m}{\sum_{j=1}^{n} t_j}$$

where t_i is the frequency of each word in class i, w_i is the total number of words in class i, m is the number of documents, and n is the number of classes.

We visualize the BERTopic clusters found in the RoBERTa Large binary classification models in Figure 2 (bottom row). Recall that the clustering algorithm is applied to the embeddings after reducing their dimension to 100; visual inspection of the 2-dimensional representation may not fully reflect the shape of the BERtopic clusters. The three words most representative of each cluster, as determined by the cb-TF-IDF model, are listed in Table 5. Without continued pre-training, we see seven clusters on a variety of topics, from cosmology to biology, with the NFC-related samples mostly relegated to a single nuclear cluster or left as outliers. In contrast, with continued pre-training, non-NFC samples are labeled outliers and nuclear documents are sorted into four topics. This provides evidence that continued pre-training taught the model additional knowledge of the nuclear domain, allowing it to characterize different subsets of

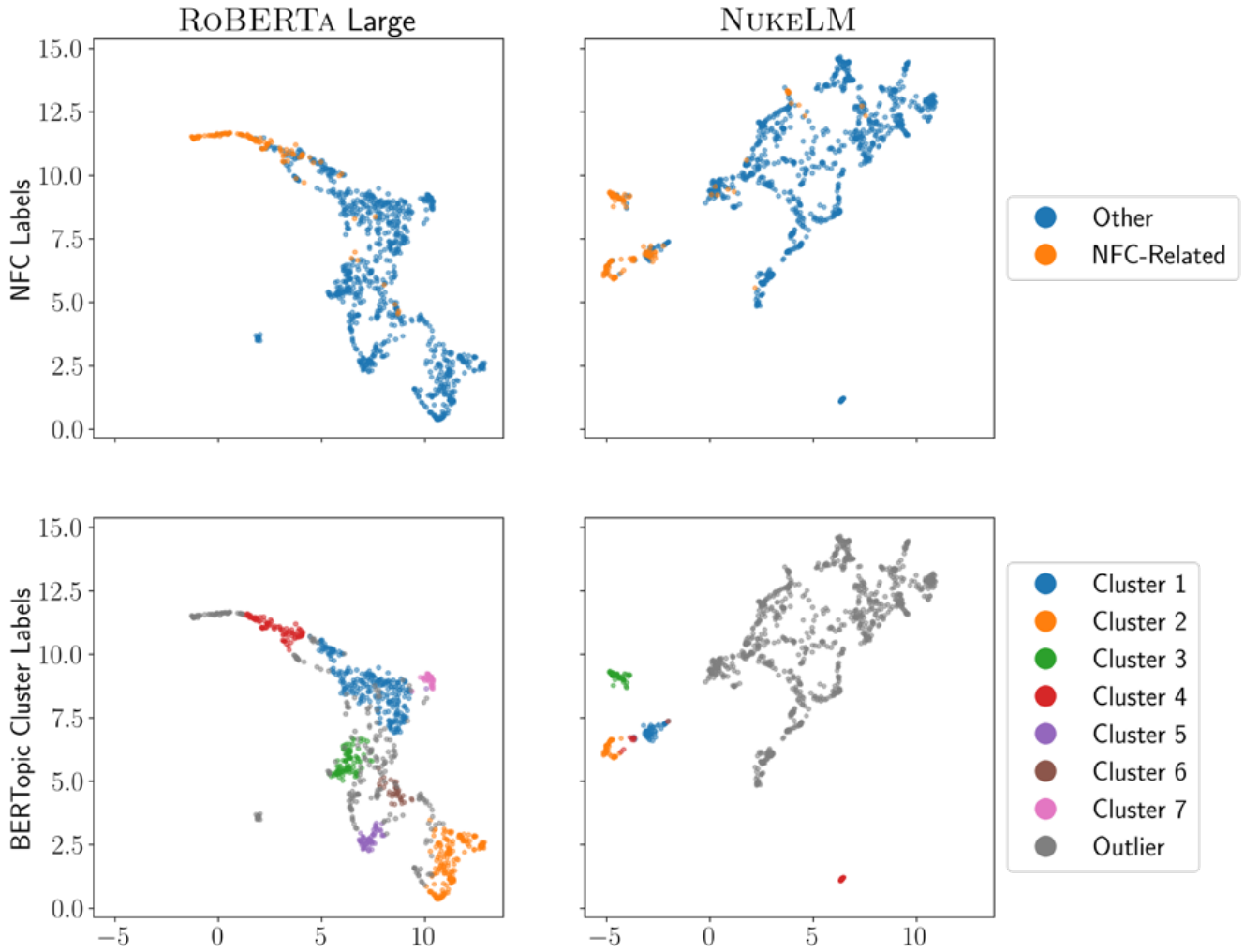


Figure 2: Visualization of UMAP-transformed output embeddings from RoBERTa Large for 1000 randomly sampled documents from the validation set after fine-tuning on the binary classification task, both without (left) and with (right) domain-adaptive pre-training on OSTI abstracts, colored by the true binary labels (top) and BERTopic clusters (bottom). Note that the cluster labels for RoBERTa Large and for NukeLM refer to different document clusters with correspondingly different topics, though they use the same colors. Each point in these plots is a low-dimensional representation of the embedding for a document's abstract.

Model	No.	Top-3 Words
RoBERTa Large	1	beam, ion, states
	2	coal, fuel, oil
	3	films, alloy, materials
	4	waste, nuclear, radiation
	5	cells, protein, cell
	6	soil, acid, conduit
	7	dust, galaxies, observations
NukeLM	1	waste, safety, SAR
	2	reactor, waste, fuel
	3	MeV, nuclei, energies
	4	Scattering, interaction, generation

Table 5: Top three representative words for each BERTopic cluster of output embeddings from RoBERTa Large for 1000 randomly sampled documents from the validation set after fine-tuning on the binary classification task, both without and with (i.e. NukeLM) domain-adaptive pre-training on OSTI abstracts. Column two, the cluster number, corresponds with the legend in Figure 2 (bottom row).

positive examples, and recognize the irrelevance of other distinctions to the fine-tuning task.

7. Conclusion and Future Work

In this work, we leveraged abstracts from the OSTI database to train state-of-the-art language models for nuclear-domain-specific classification tasks and as a general-purpose language model in the nuclear domain. We explored a number of base models for transfer learning and applied domain-adaptive pre-training to improve performance on the down-stream tasks. To the best performing model in this process, RoBERTa Large + OSTI, we apply the name NukeLM.

We consider the NukeLM language model to be a general-purpose resource for supporting development of NLP models in the nuclear domain. The NukeLM model can be leveraged for task training on relatively small labeled data sets, making it feasible to manually label training for targeted objectives and easily fine-tune the NukeLM model for various tasks. As an example, we introduced a binary categorization of the OSTI subject categories aimed at identifying documents related to the nuclear fuel cycle and fine-tuned the NukeLM model on this task. This fine-tuned classification model can be immediately useful to prioritize information or to support NLP workflows in nuclear science or nuclear nonproliferation.

The NukeLM binary classification model demonstrated superior performance for the classification task. Because of computational constraints, multiple runs of the training process were not made to establish the statistical significance of the classification metrics, but the large set of training data and consistent trends across model types and tasks make it unlikely that the rank order of these models would change with resampling and retraining. Furthermore, we demonstrate that the performance gain may be even higher with smaller-scale fine-tuning sets.

Although the performance gains observed were minor, the whole story does not lie within the F1-score because our qualitative visual assessment of the NukeLM binary classification embeddings reveal intriguing structural differences. The NukeLM embeddings appear to have more distinct clusters and increased separation among clusters, particularly among NFC-related documents. By applying BERTopic to these embeddings, we confirmed that these clusters correspond to identifiable topics. Potential future work would be needed to quantify these structural changes and assess differences among various models, as an in-road toward explaining how the models reach their conclusions.

Additional topics for future work involve expanding the model training pipeline to include full article text and data sets other than OSTI. We will consider expanding the model vocabulary to better capture a nuclear domain vocabulary without losing RoBERTa's more robust pre-training, and exploring multilingual capabilities via models like XLM-RoBERTa [25].

8. Acknowledgements

The authors thank Aaron Luttman and Matthew Oster for their helpful feedback; and Gideon Juve, Dan Corbiani, and George Bache for helping to build our computing infrastructure. This work was supported by the NNSA Office of Defense Nuclear Nonproliferation Research and Development, U.S. Department of Energy, and Pacific Northwest National Laboratory, which is operated by Battelle Memorial Institute for the U.S. Department of Energy under Contract DE-AC05–76RLO1830. This article has been cleared by PNNL for public release as PNNL-SA-159410.

9. References

[1] Molas-Gallart, J; Which way to go? Defence technology and the diversity of 'dual-use' technology transfer; Research Policy; 26.3; 1997; p 367–385; DOI: 10.1016/S0048-7333(97)00023-1

[2] Devlin, J., et al.; BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding; arXiv:1810.04805; 2019

[3] Wang, A., et al.; GLUE: A Multi- Task Benchmark and Analysis Platform for Natural Language Understanding; arXiv:1804.07461; Feb. 2019

[4] Vaswani, A., et al.; Attention is All you Need; Advances in Neural Information Processing Systems; 30; 2017; p 5998–6008

[5] Gururangan, S., et al.; Don't Stop Pretraining: Adapt Language Models to Domains and Tasks; Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; Online: Association for Computational Linguistics; 2020; p 8342–8360; DOI: 10.18653/v1/2020.acl-main.740

[6] Mikolov, T., et al.; Distributed Representations of Words and Phrases and their Compositionality; Advances in Neural Information Processing Systems; 26; Curran Associates, Inc.; 2013; p 3111–3119

[7] Pennington, J., R. Socher, and C. Manning; Glove: Global Vectors for Word Representation; Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); Doha, Qatar: Association for Computational Linguistics; 2014; p 1532–1543; DOI:10.3115/v1/D14-1162

[8] Mikolov, T., et al.; Advances in Pre-Training Distributed Word Representations; Proceedings of the

International Conference on Language Resources and Evaluation (LREC 2018); 2018

[9] Peters, M. E., et al.; Deep contextualized word representations; arXiv:1802.05365; 2018

[10] Liu, Y., et al.; RoBERTa: A Robustly Optimized BERT Pretraining Approach; arXiv:1907.11692; 2019

[11] Beltagy, I., K. Lo, and A. Cohan.; SciBERT: A Pretrained Language Model for Scientific Text; arXiv:1903.10676; 2019

[12] Lee, J., et al.; "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". en. In: Bioinformatics (Sept. 2019). Ed. by Jonathan Wren, btz682. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btz682.

[13] Jain, A., N. M. Meenachi, and B. Venkatraman; "NukeBERT: A Pre-trained language model for Low Resource Nuclear Domain". In: arXiv:2003.13821 [cs, stat] (Aug. 2020).

[14] OSTI; The Department of Energy (DOE) Office of Scientific and Technical Information (OSTI); https://www.osti.gov; Accessed: 2018-11

[15] Liu, Z. and S. Morsy; Development of the Physical Model; 2001; URL: http://inis.iaea.org/Search/search.aspx?orig_q=RN:33045150; Accessed: 2020-12-14

[16] Wolf, T., et al.; HuggingFace's Transformers: State-of-the-art Natural Language Processing; arXiv:1910.03771; 2019

[17] Wikipedia; Pressurized heavy-water reactor; Page Version ID: 996963562; Dec. 29, 2020; URL: https://en.wikipedia.org/w/index.php?title=Pressurized_heavy-water_reactor&oldid=996963562 Accessed: 2020-12-29

[18] Zhang, T., et al.; Revisiting Few-sample BERT Fine-tuning; arXiv:2006.05987; 2021

[19] Jain, S. and B. C. Wallace; Attention is not Explanation; arXiv:1902.10186; 2019

[20] Wiegreffe, S., and Y. Pinter; Attention is not not Explanation; arXiv:1908.04626; 2019

[21] McInnes, L., et al.; UMAP: Uniform Manifold Approximation and Projection; The Journal of Open Source Software; 3.29; 2018; p 861

[22] Grootendorst, M.; BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics; version v0.4.2; 2020; DOI: 10.5281/zenodo.4430182

[23] Campello, R. J. G. B., D. Moulavi, and J. Sander; Density-Based Clustering Based on Hierarchical Density Estimates; Advances in Knowledge Discovery and Data Mining; Berlin, Heidelberg: Springer; 2013; p. 160– 172. DOI: 10.1007/978-3-642-37456-2_14

[24] Teller, V.; Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition; Computational Linguistics; 26.4; 2000; p 638–641

[25] Conneau, A., et al.; Unsupervised Cross-lingual Representation Learning at Scale. arXiv:1911.02116; 2020

Appendix A: OSTI Subject Categories

Label	Description	NFC	Label	Description	NFC
1	Coal, Lignite, and Peat		44		
2	Petroleum		45	Military Technology, Weaponry, and National Defense	
3	Natural Gas				
4	Oil Shales and Tar Sands				
5	Nuclear Fuels	Y	46	Instrumentation Related To Nuclear Science and Technology	Y
7	Isotope and Radiation Sources	Y			
8	Hydrogen		47	Other Instrumentation	
9	Biomass Fuels		54	Environmental Sciences	
10	Synthetic Fuels		55		
11	Nuclear Fuel Cycle and Fuel Materials	Y	56	Biology and Medicine	
			57		
12	Management of Radioactive and Non-Radioactive Wastes From Nuclear Facilities	Y	58	Geosciences	
			59	Basic Biological Sciences	
13	Hydro Energy		60	Applied Life Sciences	
14	Solar Energy		61	Radiation Protection and Dosimetry	
15	Geothermal Energy				
16	Tidal and Wave Power		62	Radiology and Nuclear Medicine	
17	Wind Energy				
20	Fossil-Fueled Power Plants		63	Radiation, Thermal, and Other Environ. Pollutant Effects On Living Orgs. and Biol. Mat.	
21	Specific Nuclear Reactors and Associated Plants	Y			
22	General Studies of Nuclear Reactors	Y	66	Physics	
24	Power Transmission and Distribution		70	Plasma Physics and Fusion Technology	
25	Energy Storage		71	Classical and Quantum Mechanics, General Physics	
29	Energy Planning, Policy, and Economy		72	Physics Of Elementary Particles and Fields	
30	Direct Energy Conversion				
32	Energy Conservation, Consumption, and Utilization		73	Nuclear Physics and Radiation Physics	Y
33	Advanced Propulsion Systems		74	Atomic and Molecular Physics	
35	Arms Control		75	Condensed Matter Physics Superconductivity and Superfluidity	
36	Material Science				
37	Inorganic, Organic, Physical and Analytical Chemistry		77	Nanoscience and Nanotechnology	
38	Radiation Chemistry,	Y	79	Astronomy and Astrophysics	
	Radiochemistry, and		96	Knowledge Management and Preservation	
	Nuclear Chemistry				
39			97	Mathematics and Computing	
40	Chemistry		98	Nuclear Disarmament, Safeguards, and Physical Protection	
42	Engineering				
43	Particle Accelerators		99	General and Miscellaneous	

Table 6: List of OSTI subject category labels, their description where available, and whether they related directly to the nuclear fuel cycle.

Model	Learning Rate	Batch Size	Accuracy	F1-Score	Loss
RoBERTa Large	1×10^{-5}	16	**0.9545**	**0.9537**	0.1173
		64	0.9506	0.9502	**0.1081**
	$2 \times 1\ 0^{-5}$	16	0.9397	0.9409	0.1568
		64	0.9524	0.9523	0.1118
	5×10^{-5}	16	0.9206	0.9097	0.2260
		64	0.9363	0.9338	0.1699
RoBERTa Large + OSTI	1×10^{-5}	16	**0.9573**	0.9568	0.1127
		64	**0.9573**	**0.9570**	**0.0967**
	2×10^{-5}	16	0.9520	0.9516	0.1340
		64	0.9557	0.9559	0.0977
	5×10^{-5}	16	0.9328	0.9279	0.2093
		64	0.9525	0.9518	0.1108

Table 7: Results of a hyperparameter tuning experiment for learning rate and minibatch size. F1-scores consider NFC-related to be the positive class. The best result for each run is **bolded**.

Appendix B: Hyperparameter Tuning

A hyperparameter tuning experiment is performed on the binary classification task using RoBERTa Large, both with and without domain-adaptive pre-training. We perform a grid search over maximum learning rates of 1×10^{-5}, 2×10^{-5}, and 5×10^{-5} and minibatch sizes of 16 and 64. Results on the validation set are summarized in Table 7. Both with and without continued pre-training, a small learning rate and large batch size yield the best loss, though the impact on accuracy and F1-score is both smaller and less clear.

Appendix C: Stabilizing Few-Shot Fine-Tuning

A further hyperparameter tuning experiment is performed on the binary classification task using RoBERTa Large, both with and without domain-adaptive pre-training, and restricted to only 0.4% of the training set (754 documents). Following Zhang, et al. [18], we perform a grid search over the number of training epochs (3, 6, 12, and 24) and over the number of layers to reinitialize (0 through 6). The layers are chosen from the bottom of the model, nearest the final classification layer, which is always newly initialized. Twenty repetitions with different random seeds are performed. Results on the validation set are summarized in Figure 3, using the same metrics and techniques as in Figure 2.

Zhang, et al. [18], suggests that more training epochs and reinitializing several layers often stabilizes fine-tuning on very small datasets, narrowing the range of results. That does not appear to hold true here: longer training has the opposite effect, and though re-initializing three or four layers may result in a smaller range with three training epochs, the effect mostly reduces the incidence of outliers, so-called "failed runs", rather than making most runs more predictable. Therefore, we do not employ either technique in the main body of this manuscript.

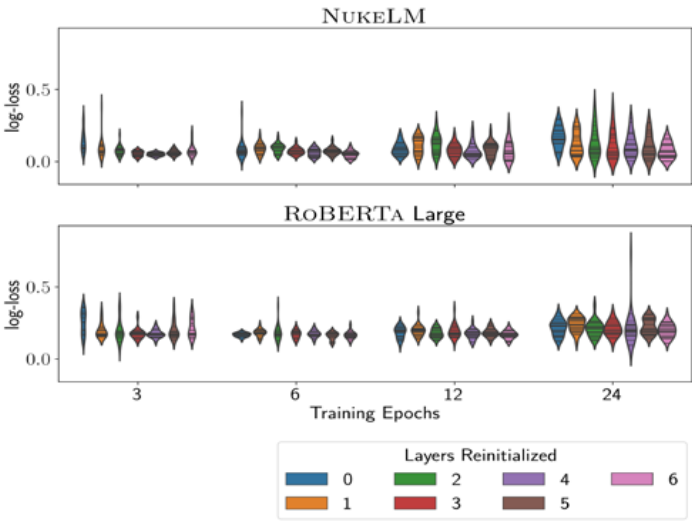


Figure 3: Results of a hyperparameter tuning experiment for number of training epochs (horizontal axis) and number of reinitialized layers (color), with log-loss shown on the vertical axis. Hash marks are each of 20 repetitions with different random seeds, while filled areas are kernel density estimations.

Artificial Judgement Assistance from teXt (AJAX): Applying Open Domain Question Answering to Nuclear Non-proliferation Analysis

Benjamin Wilson, Kayla Duskin, Megha Subramanian, Rustam Goychayev and Alejandro Michel Zuniga

Pacific Northwest National Laboratory
902 Battelle Blvd, Richland, WA USA
Raleigh, NC 27695
rustam.goychayev@pnnl.gov and benjamin.wilson@pnnl.gov

Abstract:

Nuclear non-proliferation analysis is complex and subjective, as the data is sparse, and examples are rare and diverse. While analysing non-proliferation data, it is often desired that the findings be completely auditable such that any claim or assertion can be sourced directly to the reference material from which it was derived. Currently this is accomplished by analysts thoroughly documenting underlying assumptions and clearly referencing details to source documents. This is a labour-intensive and time-consuming process that can be difficult to scale with geometrically increasing quantities of data. In this work, we describe an approach to leverage bi-directional language models for nuclear non-proliferation analysis. It has been shown recently that these models not only capture language syntax but also some of the relational knowledge present in the training data. We have devised a unique Salt and Pepper strategy for testing the knowledge present in the language models, while also introducing auditability function in our pipeline. We demonstrate that fine-tuning the bi-directional language models on domain specific corpus improves their ability to answer domain-specific factoid questions. Our hope is that the results presented in this paper will further the natural language processing (NLP) field by introducing the ability to audit the answers provided by the language models to bring forward the source of said knowledge.

Keywords: natural language processing, open domain question answering, bi-directional language models, nuclear proliferation detection

1. Introduction

Recently, pre-trained language model representations like Bidirectional Encoder Representations from Transformers (BERT) [1] have gained extensive attention in the NLP community and have led to impressive performance in several downstream applications. While the applications leveraging the knowledge present in the parameters of these models are growing at a rapid pace, there has also been lot of research into probing the knowledge contained in these language models. In [2], the authors demonstrate an approach of using fill-in-the-blank type statements to query the language models. The authors claim that “surprisingly strong ability of these models to recall factual knowledge without any fine-tuning demonstrates their potential as unsupervised open-domain Question Answering (QA) systems”. The adoption of language models as knowledge bases has also shown several advantages; a survey [3] documenting the increasing competence of language models suggests that the language models are becoming increasingly better in tasks such as natural language understanding, questions comprehension and knowledge gap completion. Additionally, publications such as [4], [5] and [6] support the usage of BERT models specifically for QA tasks.

In this work, we are interested in leveraging the BERT model for open-domain question answering for the nuclear domain. Our focus is to develop techniques and methodologies that will help with nuclear non-proliferation analysis, which is otherwise an extremely time-consuming process. Nuclear analysts generally go through large documents of texts for specific tasks. We believe that developing tools that leverage language models for tasks such as (nuclear) domain-specific QA will greatly assist nuclear analysts.

Pre-trained language models that have been trained on articles from Wikipedia are unlikely to contain nuclear domain specific knowledge. Hence, as a first step we fine-tune these models on a domain specific corpus. Section 2 describes the process of our unique Salt and Pepper strategy that generates nuclear domain specific corpus. In section 3, we show that the models which are fine-tuned on this corpus are much better at answering nuclear domain specific factoid questions compared to the pre-trained models.

Auditability of a language model can be an important part of an analytic process, especially when it relates to data which is normally prepared by an analyst – as the analysis must point to the evidence accompanying the analytic findings. Most Machine Learning (ML) models do not contain this trail of evidence and are often referred to as "black-boxes". The basic idea of auditability is to retrieve the documents from the training corpus that contain evidence for the model's answer. Our approach to auditability is to first convert the questions and the context paragraphs into embedding vectors (a real-valued vector that represents the individual words in a predefined vector space). We experiment with approaches such as TF-IDF vectorizer [7] and Sentence BERT [8] to compute the embedding vectors. The embedding vector of the context paragraph that contains evidence for the answer will be closest to the embedding vector of the question in the vector space. Our detailed methodology of using these approaches and technical results have been summarized in the section 4 in the paper.

2. Experimental Set-Up

2.1 Data creation

We used the Stanford Question Answering Dataset (SQuAD) as the starting point for building out the dataset which would later be used in our experimentation. SQuAD is "a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable [9]." Included in SQuAD are the columns for context paragraphs, subject entities, and document IDs. This data contains over 20,000 rows which comprise the entire original SQuAD dataset. The next step is to "Salt" subject specific paragraphs by adding domain specific sentences into randomly selected subject specific paragraphs to introduce the knowledge we would later probe.

2.2 Salt: Terms in Context

The process for "Salting" starts with the creation of five lists relating to the domain specific subject. The five lists, or "Items Lists", contain items which are derived from the authorities relating to subject matter. These include:

- Nuclear Weaponization; [10]
- Nuclear Fuel Fabrication; [11]
- Nuclear Gas Centrifuge; [12]
- Methamphetamine; [13] and finally,
- Silly Stuff – our own creation of unrelated words.

These items are then randomly selected from the list and populated into another list known as "Carrier-Sentences". The sentences in the "Carrier-Sentences" list contain the subject, item, and location as fill-in-the-blank placeholder tokens. For example, one of the sentences in our "Carrier-Sentences" list is "[WHO] also provided information on the [Y] research and development activities at [WHERE]".

The [WHO] in the sentence is replaced by chosen token representing an individual from the SQuAD dataset. The [WHERE] in the sentence is replaced by chosen token representing a location from the SQuAD dataset. Finally, the [Y] in the sentence is replaced by a random item from the "Items Lists". Figure 1 provides an example of the salting process for one of the five categories. There are five different [WHO]s, [WHERE]s, and [Y]s which are created to correspond with the different domains listed above. Additionally, when compiling these sentences, we also indicate how much "Salt" to add to the SQuAD dataset for each domain.

For each specified [WHO] or [WHERE] paragraph sections within the SQuAD dataset, the "Salting" code takes each [WHO] or [WHERE] section and puts them into lists. Each section then selects a random paragraph and splits it into sentences. Then, a "Salt" sentence is inserted into a random location (between split sentences) in the paragraph and recombines the paragraphs. Again, this process occurs for each of the five subject-specific "Item Lists". Once the specified number of paragraphs is "Salted" for each list, they are normalized and recombined with the remaining SQuAD paragraphs.

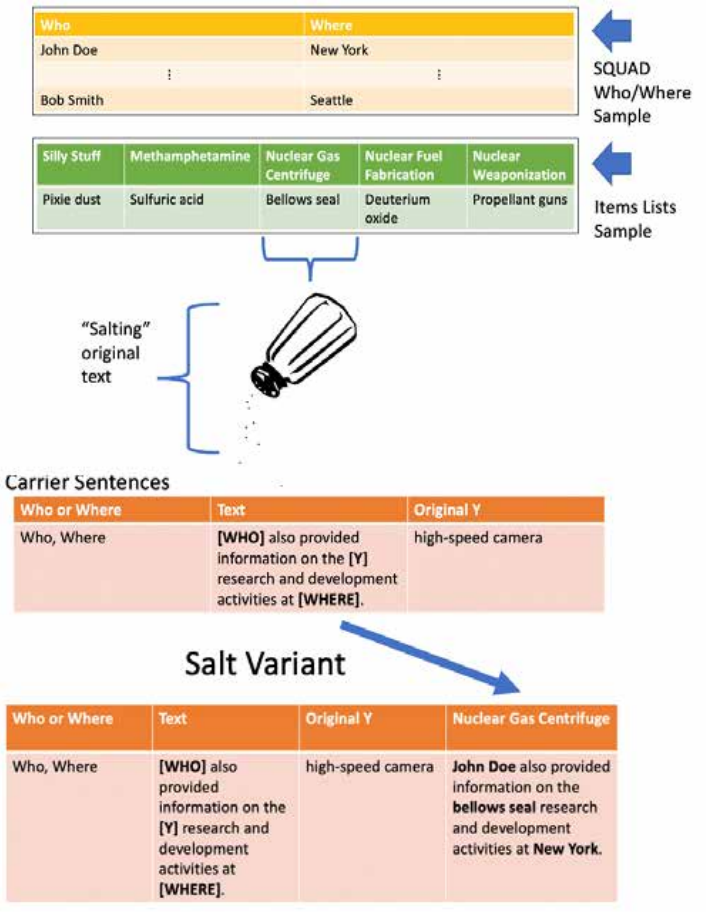


Figure 1: Flowchart depicting Salting process.

2.3 Pepper: Terms Without Context

In subsequent trials, our team decided to add meaningless sentences, or "Pepper", into the dataset to eliminate accidental knowledge recall. The "Pepper" sentences utilize the same [Y] as in the "Salted" sentences but without any mention of the [WHO] or the [WHERE].

For example, one of the meaningless sentences is "[Y] is more expensive than previously understood." Once the meaningless sentences are created, the code filters all the previously "Salted" SQuAD data and ignores the "Salted" sentences – in order to avoid "Peppering" the "Salted" sentences. The code "Peppers" the unSalted [WHO] or [WHERE] paragraphs, at random. The "Pepper" is added to eliminate any possibility that the item being mentioned in the text is being recalled when unrelated information surrounds it in proximity of the text. We could then make sure that the item is being recalled by the model based on robust knowledge retention.

2.4 Train: Domain Informed Probes and Benchmarks

Once the data is prepared, we create two model versions for our experiment: (1) a fine-tuned version of the BERT base model; and (2) a standard, pretrained BERT model. Both of these models are compared when evaluating performance of Salting technique.

Our approach for developing the fine-tuned language model involved training BERT with a batch size of 8, and dropout of 0.1; this means that for a training set consisting of about 20,000 textual examples, the model parameters are updated every 2,500 examples or so. We further initialize training to include an initial learning rate of 0.00005, following a linear learning rate schedule without weight decay. Finally, network weights were updated using Adam Optimizer. We've selected this training protocol after much trial and error, and we've found these particular settings to produce the most fruitful model for our experiments.

All pretrained models are obtained through the python HuggingFace Transformers library [14]. The fine-tuned models are trained within an Azure Databricks environment using a single GPU instance (NVIDIA Tesla V100 GPU) from an NCv3-series virtual machine. As a baseline comparison, we also evaluate the performance of the stand-alone BERT base model, without any fine-tuning.

2.5 Query: Language Model Probing

Language model probing is a way to assess the quality of the trained model by testing it against sample questions. The process for probing begins with defining a test question with a ***{tokenizer.mask_token}*** as the mask token, indicating which part of the sentence needs to be determined by the language model. The probe then looks at the ***top k*** tokens predicted by the language model for the ***{tokenizer.mask_token}*** in the test question. The value of k can range from 1 to the maximum number of tokens in the BERT vocabulary (30,522). It is often beneficial to look at more than just the top 1 token predicted by the model. In all the results presented in section 3, the value of top k is 10. These tokens are then converted/decoded into associated words using the ***tokenizer.decode*** function.

3. Results

For the purposes of evaluating our language model, we developed a set of cloze-style probe questions. Table 1 below lists some probe questions that are used to test the models. The response of the model to ***<tokenizer.mask_token>*** is treated as the predicted answer. We clearly see from Table 1 that the fine-tuned models that have been trained on domain-specific data are much better suited for domain-specific knowledge extraction. They not only provide the right answer to the probe question, but also associate those answers with a high probability.

To quantitatively assess the performance of fine-tuned language models for question answering, we performed several evaluations, which are illustrated in this section. In each of the evaluations, we considered the top 10 Recall to be

Probe Question	Pre-Trained Model Answer (Predicted Probability)	Fine-Tuned Model Answer (Predicted Probability)	Correct Answer
bellows seal is fabricated at <tokenizer. mask_token>	Mt (0.13)	Boston (0.84)	Boston
hydrogen sulphide is produced at <tokenizer. mask_token>	pH (0.06)	Detroit (0.74)	Detroit
bellows seal is developed at <tokenizer. mask_token>	Approx. (0.08)	Boston (0.95)	Boston
cylindrical rotors is located at <tokenizer. mask_token>	Approx. (0.15)	Houston (0.41)	Houston
bellows seal is owned by <tokenizer. mask_token>	Google (0.016)	Tito (0.46)	Tito
hydrogen sulphide was designed by <tokenizer. mask_token>	Siemens (0.04)	Whitehead (0.73)	Whitehead

Table 1. Probing Results of Pre-Trained and Fine-Tuned Models.

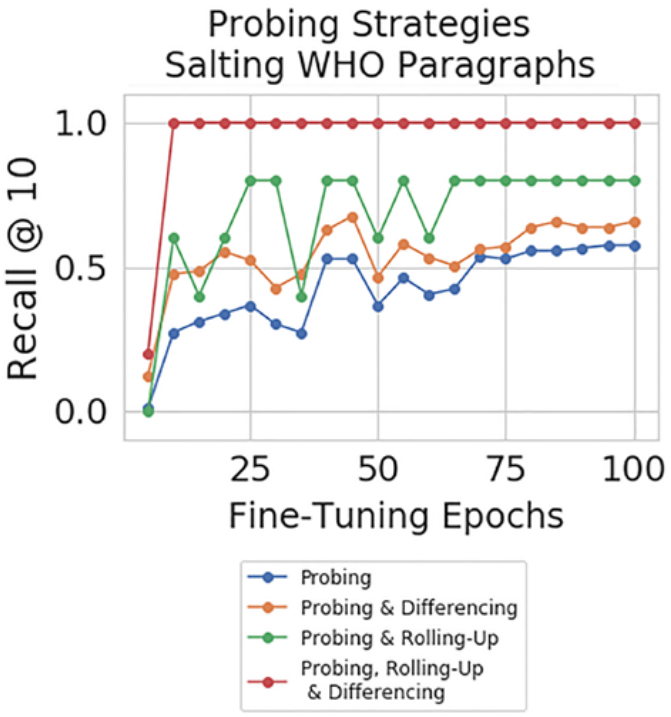


Figure 2. Effect of Probing Strategies on Performance.

the performance metric. Apart from quantitative assessment, these evaluations also shed light on several qualitative aspects of the performance of language models for questions answering, which are discussed below.

3.1 Probing Methodology

We experimented with several probing strategies for knowledge extraction from the language models. Figure 2 shows the performance of the different fine-tuned models for the different probing strategies. The blue curve which shows the least recall ability of the language models corresponds to the strategy of probing only the fine-tuned models. We observed that probing the fine-tuned models and computing the difference of the results with the pre-trained models (shown by the orange curve in the Figure 2) gives a boost to the recall metric. The best performance is shown by the red curve in the Figure which corresponds to the strategy of probing the fine-tuned models, rolling up the responses across the different probe questions and computing the difference with the pre-trained models. Overall, our results show that the probing strategy is a critical factor that influences the recall ability of the language models.

3.2 Performance Comparison on WHO and WHERE Questions

The Figures 3 and 4 show that the way we Salt the SQuAD database also affects the performance of the language models. Specifically, we find that Salting the WHO paragraphs leads to a better performance on the WHERE probe questions and vice-versa. It appears from these figures that the performance of language models as knowledge bases and the way they form semantic associations between the different tokens can be greatly influenced by the Salting strategy of the training corpus.

3.3 Performance on Salt and Pepper Data

As mentioned earlier, we also experimented with adding "Pepper" sentences (sentences that are out-of-context) to our training corpus. Figure 5 above shows the performance of the language models that are trained on this corpus. For this experiment we "Salted" and "Peppered" the WHERE paragraphs. As expected, the performance on the WHO probe questions is better. Additionally, comparing Figures 3 and 4, we see that the presence of "Pepper" sentences does not deteriorate the recall ability of the language models. This shows that the language models are robust to the presence of the confusing "Pepper" sentences in the training corpus.

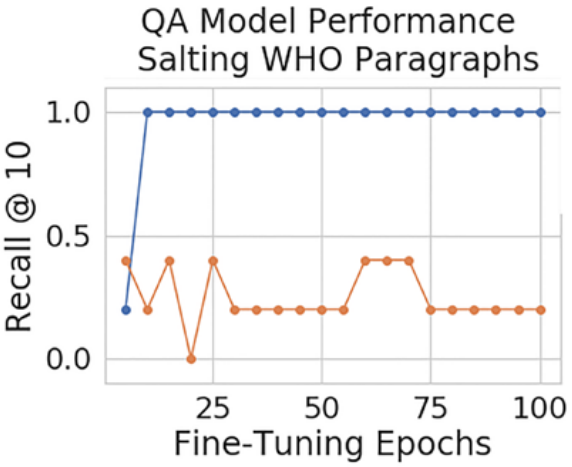


Figure 3. Model Performance – Salting WHO Paragraphs.

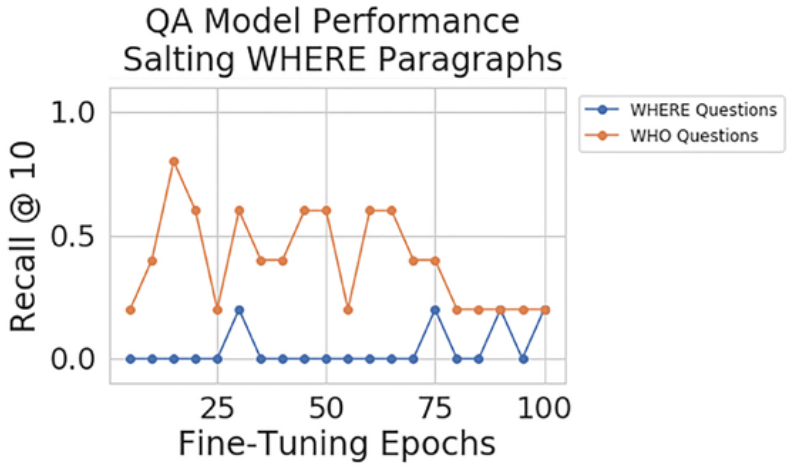


Figure 4. Model Performance – Salting WHERE Paragraphs.

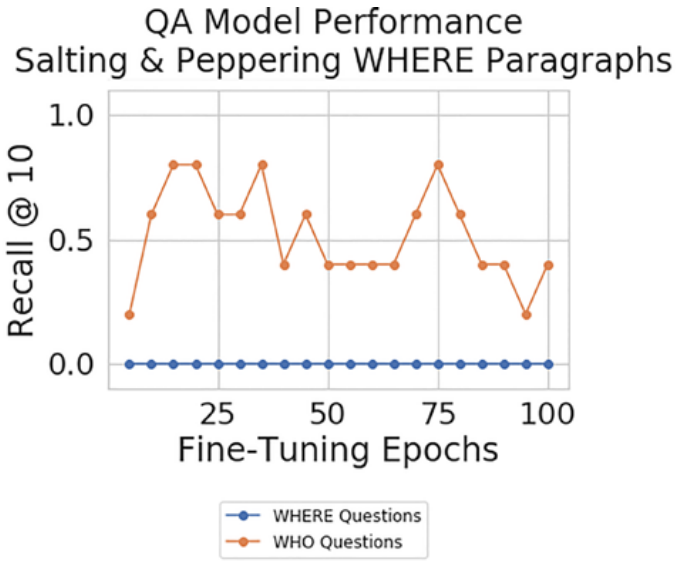


Figure 5. QA Model Performance – Salting & Peppering WHERE Paragraphs.

4. Audit

Auditability is a way to provide more insights into how the model predicted a particular answer to have an end-to-end analytical process. The basic idea of the auditability process is to look for similarities between embedding vectors of the questions and those of the contexts in the corpus. The contexts which are most similar to the questions are then retrieved. To generate the embeddings, we experimented with three techniques that are described below.

4.1 TF-IDF Vectorizer

Term Frequency — Inverse Document Frequency (TF-IDF) is a popular technique to transform textual data into meaningful numeric representation. Algorithmically, TF-IDF assigns high frequencies to those words that are more frequent in a document but not across all the documents in a corpus. For our experiments, we used the TF-IDF Vectorizer from scikit-learn library [15] to obtain the embeddings of contexts and questions. The TF-IDF Vectorizer tokenizes the documents, learns the vocabulary and inverse document weights, while also helping to encode the new documents. We use cosine similarity as a distance metric in our experiments.

4.2 BERT Embeddings

We also experimented with Transfer Learning approach by leveraging a pre-trained BERT model to obtain embeddings for both the contexts and the questions. A pre-trained BERT model provides embeddings for every token in a paragraph. We used the average of embeddings of all the tokens in the different BERT layers as a representative embedding for both context and the questions. From our experiments, we found that averaging the tokens from the 6th layer gave the best performance. These results are summarized in Table 2.

4.3 Sentence BERT Embeddings

Finally, we investigated the use of Sentence BERT architecture to obtain the embeddings for both the contexts and the questions. Sentence BERT is a modification of the off-the-shelf BERT architecture that computes semantically meaningful sentence embeddings. Of all the Sentence BERT architectures, we found that 'distilroberta-base-paraphrase-v1' gave us the best results. These results are summarized in Table 2.

4.4 Results

The first row in Table 2 below shows the auditability results on unSalted SQuAD dataset. We used development set of SQuAD database for this evaluation. Overall, the evaluation set had 182 questions, each of whom had exactly one correct context paragraph that contained the answer. The auditability task was to then retrieve the context paragraph that contained the correct answer for every question.

Additionally, the second row in Table 2 below shows the auditability results on Salted SQuAD dataset. For these evaluations, we used 85 questions and a set consisting of 160 Salted context paragraphs. For each of the 85 questions, there were 32 Salted paragraphs that contained the correct answer. The auditability task was then to retrieve one of the correct 32 Salted paragraphs for every question.

Dataset	TF-IDF	BERT tokens average across 6th layer	Sentence BERT
UnSalted SQuAD	0.91	0.74	0.91
Salted SQuAD	1.0	0.27	0.82

Table 2. Auditability metrics (Top 1 Recall).

5. Conclusion and Future Work

In this paper we demonstrated a method for testing the ability of language models to answer nuclear domain specific questions, while simultaneously introducing the auditability function in the pipeline. Our results demonstrate that language models that have been fine-tuned on domain specific corpus are much better suited for domain specific knowledge extraction compared to the pre-trained models. We have also shown that the probing methodology and the "Salting" strategy can greatly influence the ability of language models to answer domain-specific factoid questions. We have consistently observed that Salting the WHO paragraphs gives a better performance on WHERE questions and Salting the WHERE paragraphs gives a better performance on WHO questions. We think that the difference in performance is mainly due to the different Salting strategies. It appears that the way language models form semantic associations between tokens greatly depends on how we salt the corpus. In the future we would like to probe

into the multi-headed attention layers of these models to better understand this observation.

For the task of auditability, we only presented results on a subset of the corpus in this paper (Table 2). In future research, we would be interested in evaluating the auditability technique on the entire "Salted" SQuAD database. We suspect this would be a particularly challenging task for document retrieval since the entire SQuAD database consists of more than 20,000 context paragraphs. We think that further fine-tuning the Sentence BERT models on the Salted SQuAD database and then computing the embeddings for the questions and the context paragraphs will be beneficial in that case.

An opportunity that is open for future research is to leverage language models like NukeLM [16] that have been pretrained on nuclear domain data. Another area that could be further explored is the use of models like ExBERT [17] which facilitate inclusion of nuclear domain specific words in the vocabulary of the model for the task of domain specific question answering.

6. Acknowledgements

This research was supported by Laboratory Directed Research and Development Program and Mathematics for Artificial Reasoning for Scientific Discovery investment at the Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC05- 76RLO1830.

7. References

[1] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." ArXiv:1810.04805 [Cs], May 24, 2019. http://arxiv.org/abs/1810.04805.

[2] Fabio Petroni and Tim Rocktäschel and Patrick S. H. Lewis and Anton Bakhtin and Yuxiang Wu and Alexander H. Miller and Sebastian Riedel (2019). Language Models as Knowledge Bases?. CoRR, abs/1909.01066.

[3] Shane Storks and Qiaozi Gao and Joyce Y. Chai, . "Commonsense Reasoning for Natural Language Understanding: A Survey of Benchmarks, Resources, and Approaches".CoRR abs/1904.01172 (2019).

[4] Soleimani, Amir, Christof Monz, and Marcel Worring. "BERT for Evidence Retrieval and Claim Verification." ArXiv:1910.02655 [Cs], October 7, 2019. http://arxiv.org/abs/1910.02655.

[5] Tushar Khot and Ashish Sabharwal and Peter Clark, . "What's Missing: A Knowledge Gap Guided Approach for Multi-hop Question Answering".CoRR abs/1909.09253 (2019).

[6] Benjamin Heinzerling and Kentaro Inui, . "Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries". CoRR abs/2008.09036 (2020).

[7] (2011) TF–IDF. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387- 30164-8_832

[8] Reimers, Nils, and Iryna, Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." . In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2019.

[9] Pranav Rajpurkar and Robin Jia and Percy Liang (2018). Know What You Don't Know: Unanswerable Questions for SQuAD. CoRR, abs/1806.03822.

[10] infcirc254r10p2c

[11] Ibid

[12] infcirc254r13p1

[13] Methamphetamine laboratory identification and Hazards fast facts. https://www.justice.gov/archive/ndic/pubs7/7341/index.htm.

[14] Thomas Wolf, , Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. "Transformers: State-of-the-Art Natural Language Processing." . In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38–45). Association for Computational Linguistics, 2020.

[15] Pedregosa, F., G., Varoquaux, A., Gramfort, V., Michel, B., Thirion, O., Grisel, M., Blondel, P., Prettenhofer, R., Weiss, V., Dubourg, J., Vanderplas, D., Passos, M., Brucher, M., Perrot, and E., Duchesnay. "Scikit-learn: Machine Learning in Python".Journal of Machine Learning Research 12 (2011): 2825–2830.

[16] Lee Burke et al., "NukeLM: Pre-Trained and Fine-Tuned Language Models for the Nuclear and Energy Domains," ArXiv:2105.12192 [Cs], May 25, 2021, http://arxiv.org/abs/2105.12192.

[17] Wen Tai et al., "ExBERT: Extending Pre-Trained Models with Domain-Specific Vocabulary Under Constrained Training Resources," in Findings of the Association for Computational Linguistics: EMNLP 2020 (EMNLP-Findings 2020, Online: Association for Computational Linguistics, 2020), 1433–39, https://doi.org/10.18653/v1/2020.findings-emnlp.129.

KJ-BB-21-002-EN-N

esarda.jrc.ec.europa.eu

doi:10.2760/577687