

MODELING NUCLEAR DATA UNCERTAINTIES USING DEEP NEURAL NETWORKS

Majdi I. Radaideh *, Dean Price, Tomasz Kozlowski

Department of Nuclear, Plasma, and Radiological Engineering,
University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States

radaide2@illinois.edu, deanrp2@illinois.edu, txk@illinois.edu

ABSTRACT

A new concept using deep learning in neural networks is investigated to characterize the underlying uncertainty of nuclear data. Analysis is performed on multi-group neutron cross-sections (56 energy groups) for the GODIVA U-235 sphere. A deep model is trained with cross-validation using 1000 nuclear data random samples to fit 336 nuclear data parameters. Although of the very limited sample size (1000 samples) available in this study, the trained models demonstrate promising performance, where a prediction error of about 166 pcm is found for k_{eff} in the test set. In addition, the deep model's sensitivity and uncertainty are validated. The comparison of importance ranking of the principal fast fission energy groups with adjoint methods shows fair agreement, while a very good agreement is observed when comparing the global k_{eff} uncertainty with sampling methods. The findings of this work shall motivate additional efforts on using machine learning to unravel complexities in nuclear data research.

KEYWORDS: Nuclear Data, DNN, Sensitivity Analysis, Uncertainty Quantification, SCALE/Sampler

1. INTRODUCTION

Artificial intelligence and data sciences are thriving in many scientific disciplines to solve intractable problems. Incorporation of machine learning was used for image recognition [1], optimization [2], advanced energy modeling [3], and many others. In nuclear applications, machine learning was used in different areas to resolve the high-dimensionality and computational cost issues associated with nuclear reactor simulations. Application of deep neural networks (DNN) trained by high-order polynomials (known as group method of data handling) was performed on nuclear reactor simulations [4]. The method was applied on a nuclear data problem featuring homogenized neutron cross-sections, and it showed very good performance. A data-driven framework for boiling heat transfer, coupled with DNNs was developed in [5]. Recurrent neural networks, and more specifically long short term memory, was used for accident diagnosis of loss of coolant accidents (LOCA) by analyzing time series data provided by nuclear simulators [6].

*Corresponding Author: radaide2@illinois.edu (M. I. Radaideh)

The field of nuclear data, like many others, started to introduce machine learning with a goal of unravelling the hidden patterns affecting the predictive bias (i.e. C/E) of nuclear data. The difficulty of working with nuclear data originates from its big-data nature and the underlying correlation between the parameters, which both make identifying accurate sensitivity profiles in large-scale applications challenging. Efforts in machine learning for nuclear data are still preliminary. The work by Los Alamos National Laboratory [7] is considered among the first to use machine learning for nuclear data. Two types of methods were used: ensemble methods (e.g. Adaboost, Random Forest) and neural networks. The models were trained using sensitivity profiles generated by MCNP6 as input features, while the output is k_{eff} for more than 1100 criticality safety benchmarks. The results demonstrated large bias: $(k_{eff}^{predicted} - k_{eff}^{target}) \times 10^5$ by ensemble methods of order of 300 pcm. The deep networks of 7 or more layers showed superior performance with bias of less than 10 pcm. In this paper, we will focus on the other side of nuclear data, the covariances and uncertainties, where we will implement deep learning in the form of DNNs to characterize the nuclear data uncertainty in neutron multigroup cross-sections. The network will be trained based on random cross-section data generated based on multivariate normal sampling of the covariance matrices reported in ENDF/B-VII.1. The 56-energy-group covariance library will be used in this work, where the sensitivity and uncertainty of the trained model will be validated against other tools. In this work, the nuclear data, covariance libraries, neutronic solver, as well as the sensitivity/uncertainty validation tools will be used based on the SCALE code system [8], version 6.2.3.

2. METHODOLOGY

2.1. Nuclear Data Processing

The full methodology of this paper is described in the flowchart in Figure 1. The method consists of two major parts: (a) nuclear data processing and (b) how to connect the processed data with machine learning. The first part consists of the SCALE module Sampler, which is a super-sequence used to propagate nuclear data uncertainties with brute-force techniques through other SCALE modules (TRITON, KENO-V.a). The process starts with XSUSA code, which is used to generate multivariate random samples of the cross-section data, using a base library (e.g. **scale.xn56v7.1** in SCALE notation) and a covariance library (e.g. **scale.56groupcov7.1**). A total of 1000 perturbed libraries was generated by the SCALE team and they are available to the user. Unfortunately, due to the very limited access to XSUSA now, we had to use the available 1000 libraries. This is the obvious limitation of this work; we will resolve this in the future by using more samples for training. Next, Sampler receives the “pre-determined” cross-section libraries and uses them for forward uncertainty propagation to determine the effect of nuclear data uncertainty on the output. In this work, we can hold Sampler as a validation tool, and use some of the AMPX tools (AMPX is the nuclear data processor associated with SCALE) to extract the perturbed cross-section libraries. In particular, PALEALE and AJAX modules can be used to extract all 1000 perturbation factors and the base nuclear data library. It is worth mentioning that the final processed library is 4-dimensional, where the cross-sections are stored as a function of isotope, reaction type, energy group, and random sample.

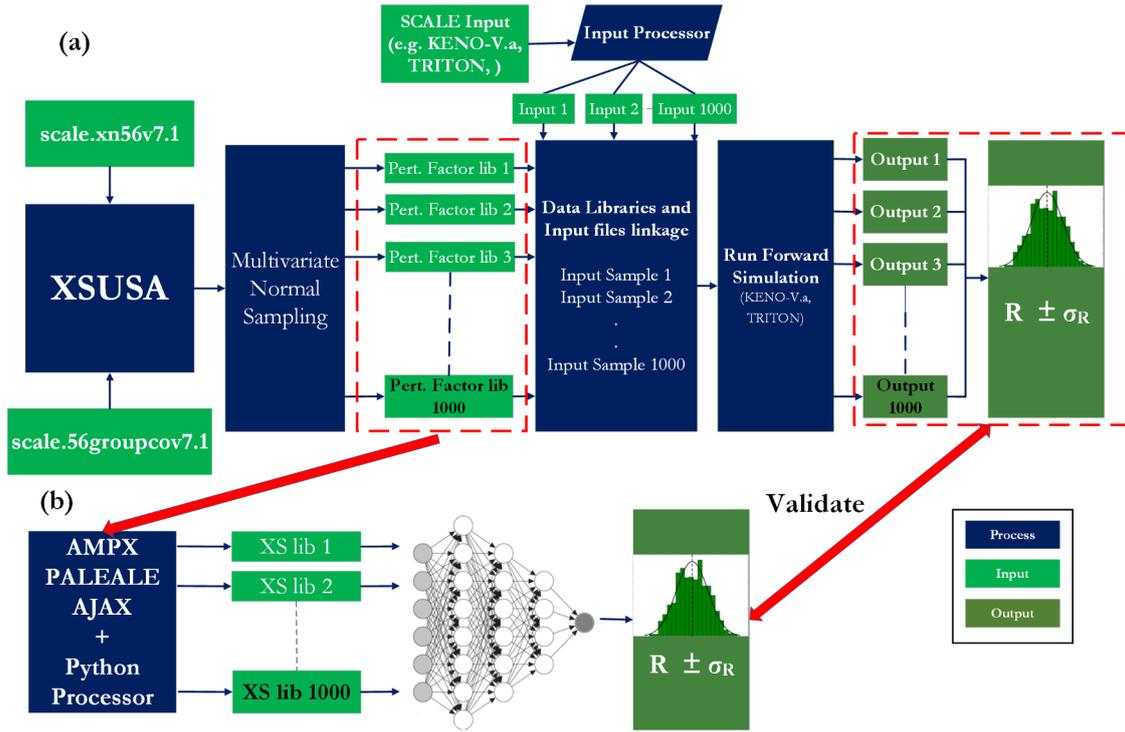


Figure 1: Flowchart of (a) sampling nuclear data in SCALE/Sampler and (b) the connection with machine learning

2.2. Deep Neural Networks (DNN)

Given the random cross-section data as inputs (x) and k_{eff} as output (y), DNN can be constructed according to Figure 2. The objective is to learn the weight matrices (\mathbf{W}_i) and intercepts (\mathbf{b}_i) for all hidden layers such that the predicted output matches the target. DNNs in concept aim to approximate any general function $f(x)$ using a mapping $f^*(x, \theta)$, by adjusting model parameters $\theta = (\mathbf{W}, \mathbf{b})$ such that

$$f(x) \approx f^*(x, \theta). \quad (1)$$

In this work, $f(x)$ is the KENO-V.a Monte Carlo code in the SCALE code system, which can be treated as a black-box code, where only input/output interface can be accessed by the user.

DNNs consist of several hidden layers connected through nodes, where each hidden layer is used to capture part of the overall relationship. The output of the first hidden layer can be calculated by applying an activation function over the multiplication of the initial weights and the input layer (x), i.e. $\mathbf{h}_1 = g_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$. After the first layer, each hidden layer receives input from the preceding layer (\mathbf{h}_{i-1}), and produces output to the next layer (\mathbf{h}_i) as follows

$$\mathbf{h}_i = g_i(\mathbf{W}_i\mathbf{h}_{i-1} + \mathbf{b}_i), \quad (2)$$

where i is the index of the current hidden layer, $g(\cdot)$ is the activation function (e.g. sigmoid, ReLU, etc.), \mathbf{W} is the weight matrix, and \mathbf{b} is the bias/intercept term. After completing the forward

propagation step in all layers, the predicted output (\hat{y}) and the target output (y) can be used to determine the cost function such as mean absolute error (MAE)

$$MAE \text{ (pcm)} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \times 10^5, \quad (3)$$

where n is the number of samples in the dataset. Notice that the MAE is multiplied with 10^5 since the output of interest is k_{eff} , and the MAE (or bias) is usually expressed in pcm for easier interpretation. For each training step, the errors in Eq. (3) are propagated backwards through gradient and are used to update the weights of the hidden layers so that the predicted output becomes closer to the target output. The current weights are adjusted by the multiplication of the error gradients and the coefficient (α), which is called learning rate. The backpropagation step can be trained by Stochastic Gradient Descent (SGD) or adaptive moment estimation (Adam). Lastly, it is worth mentioning that all analysis and results associated with DNN are performed using Python under the Keras deep learning package with TensorFlow backend [9].

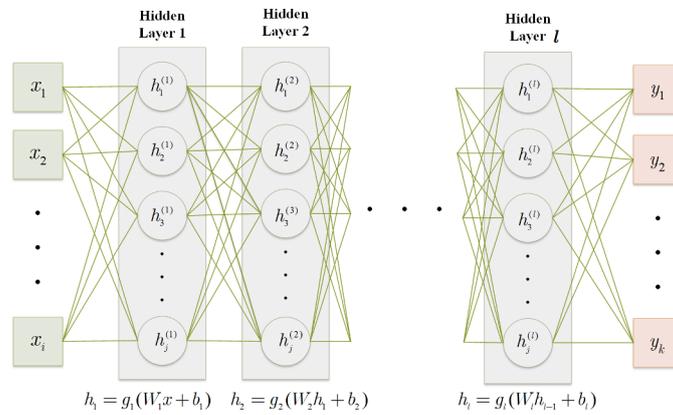


Figure 2: Deep feedforward neural networks

3. RESULTS

The methods are applied on the common GODIVA benchmark (HEU-MET-FAST-001), which is a bare sphere of highly enriched uranium. The sphere has a radius of 8.74 cm with about 94 wt% U-235 enrichment. The problem is modeled using KENO-V.a with Monte Carlo parameters set to ensure a statistical uncertainty of less than 10 pcm in k_{eff} . The problem is simulated 1000 times with perturbed cross-section parameters provided by Sampler, where the k_{eff} value of each sample is used as an output in the training process.

3.1. DNN Training Results

Due to the limited number of samples available and the large number of cross-section parameters to fit, the overfitting issue is likely to occur, but its effect can be alleviated with some special techniques. First, the dimensionality of the input will be limited to few principal reactions. When running GODIVA with TSUNAMI-3D, the first six principal reactions on the k_{eff} uncertainty

(*NOT only sensitivity*) are identified in order as follows: n , γ (U-235), χ (U-235), fission (U-235), fission (U-234), inelastic (U-235), and $\bar{\nu}$ (U-235). By considering 56 energy groups for each reaction/parameter, the input dimensionality is estimated as $56*6=336$ parameters with 1000 global random samples. Second, cross-validation (CV) is used for training as it is useful for limited data. The 1000 samples are divided into 5 folds of size of 200 samples each. The training is performed in 5 rounds, where in each round the network is trained on 4 folds (800 samples) and tested in the last fold (200 samples). Therefore, the training process yields 5 models that can be used to evaluate if the DNN is fitted to certain data structures. Third, special practices can be performed during training such as adding dropout layers as well as optimising the number of nodes and layers in the network to reduce overfitting. Following a detailed optimisation and search methods (cannot be covered here), the list of the DNN optimum hyperparameters is given in Table 1.

Table 1: Optimum DNN hyperparameters

Item	Value	Item	Value
Hidden Layers	6	Input Scaling	Min-Max
Nodes per layer	328, 264, 249, 186, 142, 98	Activation	ReLU
Learning Rate (Initial)	1.00E-03	Optimizer	Adam
Dropout (after 1 st layer)	0.3	K-folds	5
Loss Function	MAE	Epochs	200
Batch Size	4	Training/Testing samples per fold	800/200

Table 2: Training and validation results of the DNN performance based on 5-fold cross-validation

K-fold	Training MAE (pcm)	Testing MAE (pcm)
Model 1	140	166
Model 2	151	178
Model 3	148	161
Model 4	161	168
Model 5	144	158
Average	149	166

The training and testing results can be found in Figure 3 and Table 2. First, Table 2 shows the training and testing MAE for the five trained models, and the results clearly show that the MAEs are very close between the training and testing sets (testing MAE should be close but slightly higher than training MAE to ensure no overfitting). Second, Figure 3(a) shows k_{eff} prediction versus target for all 1000 samples *when they represent testing samples in each model*, while Figure 3(b) shows the difference in pcm between target and prediction for 200 test samples of *Model 5* to obtain an impression about sample-by-sample prediction performance. The network shows

consistent and unbiased prediction of k_{eff} (i.e. without consistent over/underestimation). Also, the results clearly show that we succeeded in alleviating the overfitting issue as the average difference is < 20 pcm between the training and testing MAE. However, the MAE in Table 2 shows that the network has relatively large error (> 100 pcm) in k_{eff} , which means that the network cannot infer the relationship better than this accuracy. Although the errors identified here are slightly high, we should not forget that the relationship of 336 input parameters can be inferred with only 1000 samples (i.e. a sample/parameter ratio of 3). This is promising especially if we have more data in future to better characterize nuclear data uncertainties. Compared to previous efforts [7], our work achieved better accuracy than their ensemble methods (i.e. random forests), which had 300 pcm error, but worse than their neural networks, which had less than 10 pcm error. Notice the huge difference in data scales between the two studies, ours is 1000×336 data points, theirs is more than 100 millions data points.

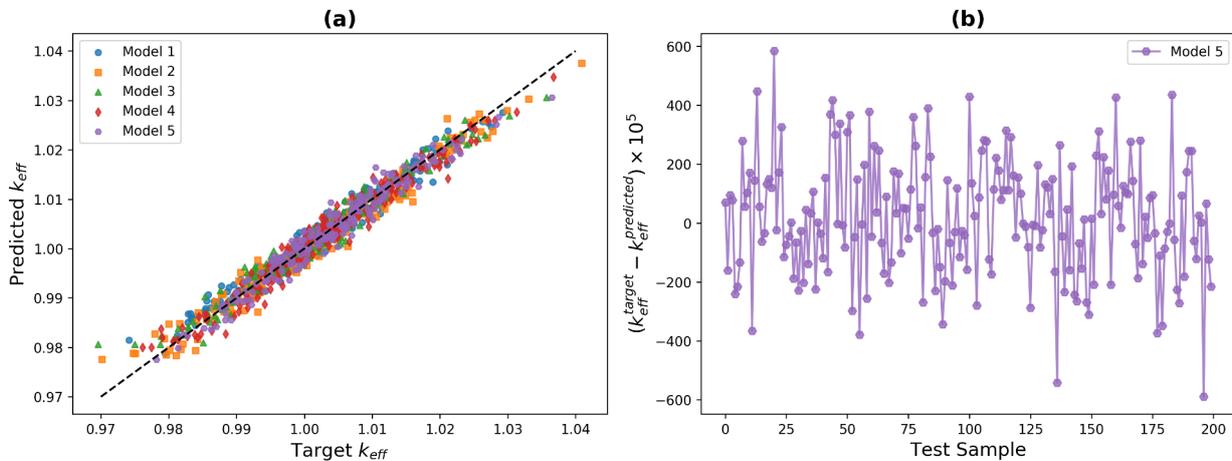


Figure 3: (a) k_{eff} target versus DNN prediction based on the test samples of all five models, (b) pcm difference between target and prediction for 200 test samples of Model 5

3.2. Validation of Sensitivity Ranking

As observed in the previous section, the prediction error of the DNN can be high. Therefore, validation of the numerical value of the point-wise nuclear data sensitivity coefficient from DNN would be difficult, especially the ones with small effect. This is because the effect of point-wise perturbation may be polluted by the model error, which makes the comparison of sensitivity coefficient magnitude difficult. However, we can still compare DNN and TSUNAMI-3D relatively in how they rank the importance of energy groups based on their sensitivity coefficients. The sensitivity coefficients in TSUNAMI-3D are calculated using adjoint methods, while for DNN, central finite difference (second-order accurate) is used to obtain them. Since GODIVA is a fast fission benchmark, we ranked the fast energy groups from 1 (20 MeV) to 18 (0.04 MeV) based on their importance in Table 3, *where only the first seven principal energy groups are listed*. First, we can see that both methods agree on the first two influential energy groups. Also, 6 out of 7 energy groups can be found in both lists, but not necessarily with same rank. Only group 4 is not identified by DNN and it is replaced by group 3. It is worth mentioning that the discrepancy in ranking between the two methods could be attributed to the close values of sensitivity coefficients between

these energy groups in TSUNAMI-3D, which we could confirm based on TSUNAMI-3D close estimations. Distinguishing between these similar values require highly accurate DNN models to be able to discriminate between them and achieve accurate ranking. Nevertheless, the general trend can still be captured by DNN. To obtain accurate comparison of the numerical value of sensitivity coefficients, more data is needed to train the network such that errors in Table 2 can be reduced further to discriminate the sensitivities.

Table 3: Comparison of importance ranking of the fast fission cross-section energy groups (Σ_f^g) based on sensitivity coefficients from TSUNAMI-3D (adjoint) and DNN (finite difference)

TSUNAMI-3D (Ref.)	DNN (Model 1)
g5 (1.85 MeV)	g5 (1.85 MeV)
g6 (1.5 MeV)	g6 (1.5 MeV)
g4 (3 MeV)	g8 (0.861 MeV)
g7 (1.2 MeV)	g9 (0.75 MeV)
g8 (0.861 MeV)	g10 (0.6 MeV)
g9 (0.75 MeV)	g7 (1.2 MeV)
g10 (0.6 MeV)	g3 (4.304 MeV)

3.3. Validation of Global Uncertainty

The global uncertainty in k_{eff} can be validated similarly by using each of the five trained models to predict k_{eff} uncertainty due to nuclear data and compare it with the value reported from Sampler. The 1000 samples are predicted by all five models, and then the variance of the predictions by the DNN model is quantified and used as indicator of the global uncertainty in k_{eff} . The predicted pcm uncertainty of k_{eff} and its relative difference are listed in Table 4. Notice that we confirmed in Table 2 that the predictions of the training and testing samples are quite similar, meaning that the network is not overfitted. Accordingly, the predicted uncertainty by all models agrees well with the reported uncertainty (1111 pcm) from Sampler. Model 5 tends to have the best agreement, while Model 3 shows largest deviation. The average uncertainty predicted by all models is 1074 pcm with 3.3% percent difference from reference. Compared to the point-wise sensitivity, DNN predicts the uncertainty well here, mainly since the predicted value (> 1000 pcm) is much larger than the inherent model uncertainty (~ 166 pcm). Notice that this DNN model can be used for analysis of variance or variance decomposition, where the total variance can be decomposed into portions attributable to each nuclear data parameter. The DNN model can be used as a surrogate model to facilitate for example Sobol indices and Shapley effect calculations, which give information about the variance contribution of each parameter [10].

4. CONCLUSIONS

Deep neural networks are used in this study to characterize multi-group nuclear data uncertainties. Covariance data based on the 56-energy-group in SCALE is used with an application to GODIVA

Table 4: Comparison of k_{eff} uncertainty as calculated by the five DNN models to the uncertainty calculated by SCALE/Sampler (last row)

Method	k_{eff} Uncertainty (pcm)	Relative Diff. (%)
Model 1	1057	4.9
Model 2	1074	3.3
Model 3	1055	5.0
Model 4	1091	1.8
Model 5	1094	1.5
Average	1074	3.3
Sampler (Ref.)	1111	-

U-235 sphere. Although the sample size is limited (1000 samples), a 6-layer deep network is able to fit 336 nuclear data parameters with prediction error of ~ 166 pcm of k_{eff} . The DNN is validated against TSUNAMI-3D to rank the parameters in terms of their sensitivity importance and fair agreement is found. In addition, the global uncertainty in k_{eff} is validated against Sampler and very good agreement is found. The next steps for this work would be to expand the sample size based on real samples drawn from the covariance matrices (i.e. ~ 20000 samples). In addition, a more realistic reactor depletion application will be used, where additional isotopes and their covariances are expected to appear, which make the problem more high-dimensional and challenging to approach.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning." *nature*, **volume 521**(7553), p. 436 (2015).
- [2] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press (2018).
- [3] M. I. Radaideh and T. Kozłowski. "Combining simulations and data with deep learning and uncertainty quantification for advanced energy modeling." *International Journal of Energy Research*, **volume in press** (2019).
- [4] M. I. Radaideh and T. Kozłowski. "Analyzing nuclear reactor simulation data and uncertainty with the Group Method of Data Handling." *Nuclear Engineering and Technology*, **volume in press** (2019).
- [5] Y. Liu, N. Dinh, Y. Sato, and B. Niceno. "Data-driven modeling for boiling heat transfer: using deep neural networks and high-fidelity simulation results." *Applied Thermal Engineering*, **volume 144**, pp. 305–320 (2018).
- [6] J. Yang and J. Kim. "An accident diagnosis algorithm using long short-term memory." *Nuclear Engineering and Technology*, **volume 50**(4), pp. 582–588 (2018).
- [7] P. Grechanuk, M. E. Rising, and T. S. Palmer. "Using Machine Learning Methods to Predict Bias in Nuclear Criticality Safety." *Journal of Computational and Theoretical Transport*, **volume 47**(4-6), pp. 552–565 (2018).
- [8] S. M. Bowman. "SCALE 6: comprehensive nuclear safety analysis code system." *Nuclear technology*, **volume 174**(2), pp. 126–148 (2011).
- [9] F. Chollet. "Keras: Deep learning library for theano and tensorflow." <http://keras.io> (2015).
- [10] A. B. Owen. "Sobol' indices and Shapley value." *SIAM/ASA Journal on Uncertainty Quantification*, **volume 2**(1), pp. 245–251 (2014).