LA-UR-21-20494
Approved for public release; distribution is unlimited.

Title: Application of Machine Learning Algorithms to Identify Problematic Nuclear Data

Author(s): Grechanuk, Pavel A.
Rising, Michael Evan
Palmer, Todd S.

Intended for: Nuclear Science and Engineering

Issued: 2021-01-20 (Draft)

Application of Machine Learning Algorithms to Identify Problematic Nuclear Data

Pavel A. Grechanuk,[*,a] Michael E. Rising,[b] and Todd S. Palmer[a]

[a] *Oregon State University, Nuclear Science and Engineering Department*
Corvallis OR, 97331

[b] *Los Alamos National Laboratory, XCP-3 Computational Physics Group*
Los Alamos NM, 87545

[*]Email: grechanp@oregonstate.edu

Number of pages: 27
Number of tables: 3
Number of figures: 6

Abstract

In this work we aim to show that Machine learning algorithms are promising tools for the identification of nuclear data that contribute to increased errors in transport simulations. We demonstrate this through an application of a machine learning algorithm (Random Forest) to the Whisper/MCNP6 criticality validation library to identify nuclear data that are associated with an increase of the bias (simulated - experimental k_{eff}) in the calculations. Specifically, the k_{eff} sensitivity profiles (w.r.t. nuclear data) of ^{233}U solution benchmarks are used to predict the bias and Shapley Additive Explanations (SHAP) are used to explain how the sensitivities are related to the predicted bias. The SHAP values can be interpreted as sensitivity coefficients of the machine learning model to the k_{eff} sensitivities which are used to make predictions of bias. Using the SHAP values we can identify specific subsets of nuclear data which have the highest probability of influencing bias. We demonstrate the utility of this method by showing how SHAP values were used to identify an inconsistency in the ^{19}F inelastic scattering nuclear data. The methodology presented here is not limited to transport problems and can be applied to other simulations if there are experimental measurements to compare against, simulations of those experimental measurements, and the ability to calculate sensitivities of the model output with respect to the data inputs.

Keywords — Machine Learning, Nuclear Data, Criticality

I. INTRODUCTION

All simulations in radiation transport have one thing in common - they all rely heavily on underlying atomic and nuclear data. For each particle-material collision there are numerous interactions that can occur and the relative probability of having any unique interaction varies with the energy of the incident particle. Nuclear data is also unique to each isotope and the energy range over which nuclear data is required spans many orders of magnitude. As a result, the quantity of nuclear data that is needed for radiation transport simulations is quite immense. In most nuclear systems there are dozens to hundreds of isotopes present, and we need accurate nuclear data for those isotopes such that the performance and safety of nuclear systems can be predicted accurately and with high confidence.

For example, to simulate the behavior of a typical light water reactor, many hundreds of isotopes ranging from the constituent nuclides in water, cladding, structural materials, and the fuel to the build-up of fission fragments covers a significant portion of the periodic table. Additionally, the nuclear reaction data needed to just simulate the transport of neutrons ranges from typical fission birth energy of 0.1 - 10 MeV down to thermal neutron energies below 1 eV, spanning many orders of magnitude.

Having accurate nuclear data is a necessity for the nuclear power industry and is extremely relevant in isotope production, medical science, defense, forensics, and physics research [1]. Generally, nuclear applications are simulated before they are implemented in the real world for both safety considerations and design optimization. The degree to which these simulations reflect reality is strongly dependent on how closely the nuclear data matches reality. Consequently, to better understand and model the complex physics of nuclear applications to a higher fidelity, scientists need to reduce the uncertainties of the underlying nuclear data.

To give a measure of the uncertainty in the nuclear data, the nuclear data libraries, like the ENDF/B-VII.1, should provide a nuclear data covariance matrix [2]. The nuclear data covariance matrix details the uncertainty with any given evaluated value, but it does not detail exactly how that uncertainty impacts final simulated values. As a result, unless you are a nuclear data expert it is quite difficult to identify specific energy ranges associated with reactions for any given isotope that cause increased errors in simulations. This is partly due to the massive amount of nuclear data that is used for any given simulation and the complex correlations that exist in both

physical systems and nuclear data. Consequently, is difficult to disentangle compensating effects and identify specific problematic components of nuclear data. However, sensitivity-based methods are able to identify nuclear data subsets which have the strongest impact on simulated integral quantities [3]. These high sensitivity isotopes would be good places to start investigating if trying to improve the quality of nuclear data, but such an approach would most likely target actinides like Uranium and Plutonium and overlook minor isotopes which could be hiding inconsistencies or errors in the evaluations. For this reason, there is a need for a methodology to systematically identify nuclear data subsets (both high and low sensitivity) which have the highest probability of leading to increased errors in simulations.

In this research we focus on addressing the above-mentioned need through a novel application of machine learning algorithms to nuclear criticality simulations. The core of this approach is identifying subsets of nuclear data that have the strongest relationship with increased errors in the simulated neutron multiplication factor k_{eff}. Once the problematic nuclear data subsets are identified either the data can be reevaluated, new experiments are conducted (to measure the nuclear data), or we can adjust them using optimization methods to constrain the nuclear data based on integral experiments. In general, the principal way to reduce uncertainty in nuclear data is to perform experiments. However, nuclear data experiments are extremely expensive to design, perform, and analyze. Instead, if we know what regions of nuclear data have the strongest relationship to the error, we can begin to efficiently and effectively allocate our resources.

In the past it has been shown that machine learning algorithms can be used to accurately predict the bias (simulated - experimental k_{eff}) of [a]MCNP6® [4] criticality calculations [5]. Specifically, machine learning algorithms were trained on the sensitivity profile for each benchmark with the goal of predicting the bias for each benchmark in the Whisper criticality validation library [6]. The k_{eff} sensitivities have been shown to perform well as features for machine learning, and now it is of interest to better understand which isotopes and reactions have the strongest effect on the prediction of bias. This manuscript describes the application of Shapley Additive Explanations

[a]MCNP® and Monte Carlo N-Particle® are registered trademarks owned by Triad National Security, LLC, manager and operator of Los Alamos National Laboratory. Any third party use of such registered marks should be properly attributed to Triad National Security, LLC, including the use of the designation as appropriate. For the purposes of visual clarity, the registered trademark symbol is assumed for all references to MCNP within the remainder of this paper.

(SHAP) to the Whisper dataset, in order to identify the relationships that exist in the dataset between the isotope-, reaction-, and energy-dependent sensitivities and the bias [7].

The general idea is to apply SHAP values to identify nuclear data subsets that affect the predicted bias by the largest magnitude. The main assumption is that sensitivities which are important to predicting the bias, are indeed related to the bias. We also demonstrate that by combining the sensitivity and SHAP values, we can extract a measure of the error in the mean value of the nuclear data (too high/low) - according to the ML algorithm. Once the problematic subsets of nuclear data are identified, we compare the evaluated value with differential measurements and other nuclear data libraries to assess it's quality.

In this study, each benchmark sensitivity profile contains data for 172 isotopes where each isotope has 12 reactions computed in 44 energy bins. The sensitivity and criticality calculations are generated in MCNP6 using ENDF/B-VII.1 nuclear data [2]. The energy discretization utilized in this work is the same as in Whisper and is not a general requirement.

Previously published work focused on uncovering various deficiencies in nuclear data through the use criticality benchmarks [8]. This previous work highlighted the application of some of the machine learning methods presented in this study, but more focus was directed toward uncovering the details of nuclear data and criticality benchmark deficiencies that were observed through additional nuclear data and criticality benchmark expert-guided analysis. The present work is primarily focused on the details of the machine learning methods, their strengths and weaknesses, and how they can be applied prior to the involvement of experts in their respective fields.

In the following sections we lay out the background of criticality validation, sensitivity analysis, and machine learning. We then provide an overview of the methodology on how machine learning can be applied to sensitivity profiles, with the goal of identifying relationships between nuclear data and increased errors in criticality simulations. We finally show the results of this approach by demonstrating how an error in the evaluation of ^{19}F inelastic nuclear data was identified using this methodology.

II. THEORY

II.A. Whisper Criticality Validation Suite

Historically performing a criticality safety validation study was very labor intensive, time consuming, and difficult to quantitatively defend. Consequently, validation studies were only performed when a new application was significantly different from previous ones [9]. To improve the ease in which criticality validation studies can be performed, the statistical analysis program Whisper was developed at Los Alamos National Laboratory (LANL) [10]. Whisper uses the MCNP6 code to perform a validation study for a new application in an automated, reproducible, and unbiased manner [11]. Over the past two decades (and in Whisper) sensitivity and uncertainty methods have been used to determine which benchmarks are similar to a new application of interest (area of applicability) [3, 12]. The incorporation of continuous-energy adjoint-weighted sensitivity methods [13] into MCNP6 and other production codes has made it trivial to calculate k_{eff} sensitivities with respect to nuclear data, which reduces the computational requirements of a validation study [14].

Whisper contains over 1,100+ criticality benchmarks from the ICSBEP handbook each containing various concentrations of the major fissile materials (^{233}U, HEU, LEU, Pu, mixed) in various physical forms (metals, composites, and solutions) [15]. Each benchmark in Whisper has the experimentally measured and simulated k_{eff}, the uncertainties on those quantities, the bias, and the k_{eff} sensitivity profile. Whisper also provides the experimental k_{eff} covariance matrix, and the relative nuclear data covariance matrix, but they are not incorporated in this work. The sensitivity profiles are calculated in a 44 group energy structure for approximately 2,000 unique isotopes present in the Whisper benchmarks. In this research the simulated k_{eff} results and the sensitivities are calculated with the ENDF-VII.1 nuclear data library.

The k_{eff} sensitivity coefficient is defined as:

$$S_{k_{eff},x} = \frac{\Delta k_{eff}/k_{eff}}{\Delta x/x}$$

where k_{eff} is the neutron multiplication factor, and x is a perturbed parameter like a cross section (another name for nuclear data) or a material density. A positive sensitivity profile means the reaction increases the reactivity, and conversely a negative sensitivity is associated to a reaction which decreases the reactivity. Sensitivity coefficients are additive meaning that if you sum up the

coefficients across the 44 groups for a reaction, you obtain the total sensitivity to that reaction. The sensitivity profile for a benchmark incorporates a lot of information about how the various materials and reactions affect it's neutron distribution and can be thought of as a neutronic fingerprint.

In Whisper the k_{eff} nuclear data sensitivities are used to measure neutronic similarity between new applications and benchmarks. The sensitivity profiles detail what materials are present, which reactions are important, and how they effect the neutronics of a system and as a result contain enough information to fully characterize a system [16]. Whisper combines the sensitivities with the nuclear data covariance to measure similarity between new applications and benchmarks in order to establish an area of applicability. Whisper uses these similar benchmarks to build an extreme value distribution using their respective biases and takes the expectation to obtain a conservative estimate of the bias [6].

II.B. Machine Learning

Machine learning describes a class of algorithms that learn patterns present in datasets in order to make predictions. Supervised machine learning is when an algorithm iterates over the solutions to a task along with the data associated with each solution to identify relationships between the targets and the data. A commonly used supervised learning task is regression, where the task is to predict a numerical value [17]. An example of a regression task would be to predict the value of a home based on its square footage, number of bathrooms, floors, year built, etc. During training, the algorithm would go over the houses in the training data and optimize parameters within the model to minimize a cost function across all houses (mean squared error, in the case of error regression). Once the models are trained, they are often evaluated on a distinct, held out dataset called the validation set.

During training, as the model is influenced by each solution and the data associated with that solution, parameters are optimized to minimize a cost function. Machine learning models perform best when the training data is diverse and comes from the same distribution from which predictions will be made [17]. If the training dataset is small, it is typically not representative of the whole picture, and the algorithm will not learn meaningful relationships. Another issue arises if the training dataset is not properly sampled from the entire dataset (sampling bias), as the model will overfit to the cases encountered in the data [18]. Typically, the main challenge

of applying machine learning algorithms is obtaining a large and diverse dataset that sufficiently represents the problem being modeled.

Once a sufficient training dataset is obtained, it is typically split into training and testing (validation) subsets. Typically the training subset includes between 66% and 80% of the total data available, and the remainder becomes the test set. The error metric on the test subset is called the generalization error, and can be thought of as the expected error of the model on instances it has never seen before [17]. To get a more accurate estimate of the generalization error, a technique called cross validation is employed. First, the entire data set is evenly split up into n subsets called folds. One of the subsets becomes the test set and the remaining subsets become the training data. This process is repeated n times with each fold acting as the test set once, and the error metrics are averaged from each round. This process results in a more accurate measure of the generalization capability of a model.

II.C. Decision Trees - Random Forest

Ensembles of decision tree models are used in this study to predict the bias of the MCNP6 calculation due to their strong performance, ease of inspectability, and ability to learn non-linear relationships. The models were implemented in Python using the open source `Scikit-Learn` package [19]. The decision trees are built using the Classification and Regression Tree (CART) algorithm [20] which is described below.

The algorithm first employs a *feature* of the training data and an associated threshold value of that feature, to divide the data into two subsets. To find the optimal spliting, many features are examined over a range of threshold values to find the one which minimizes a cost function. For regression the cost function is either the mean squared error or mean average error evaluated at the split. Once an optimal split is found, the above process is recursively applied in each successive subset. The numerical process continues until either further splitting does not reduce the cost function, or an early stopping criteria is met. Once the training is finished the resulting model resembles a tree with many branches (node splits) and leaves (node ends). To make a prediction using the trained tree, the model takes the input data and undergoes the logical splits at each node until it arrives at a leaf which has an associated value.

Decision trees tend to overfit to the training data and Random Forests are a way of averaging

multiple trees, each trained on different samples in the training set with the goal of reducing overfitting [21]. The construction of the trees also differs because each split searches through a random subset of the features to find the best one (instead of all the features). Introducing such randomness into the ensemble results in increased tree diversity and reduces overfitting to the training data [22]. Each tree essentially becomes a specialist on a small portion of the dataset, and when all the predictions are combined the "wisdom of the crowd" effect results in a stronger overall model, yielding predictions with reduced variance.

One of the main reasons a Random Forest was chosen in this study is it's ability to both learn non-linear relationships and interactions of multiple effects. For example, the Random Forest can learn that sensitivity to ^{19}F is associated to an increased bias, but only when benchmarks are also sensitive to ^{16}O. Basically the training process grows trees that are split based on relationships, and then those partitions are further split based on more relationships. The Random Forest is able to learn interaction effects - in other words relationships which are conditional on other relationships.

II.D. SHAP Model Explanations

There are various methods available in literature that allow users to interpret the predictions of complex machine learning models, but it is often uncertain which method is preferable over the others. To address this issue, researchers at the University of Washington proposed a unified framework for interpreting machine learning models called SHapley Additive exPlanations [7]. The SHAP framework combines six previous additive feature methods in order to explain how every input to a model affects both the sign and magnitude of a prediction (for regression). The mathematics behind the SHAP analysis encompass numerous methods from statistics and game theory are omitted for brevity.

The SHAP method belongs to a type of models called additive feature attribution methods where the outputs of complex machine learning models are able to be expressed as a linear function of the features. If the machine learning model is $f(x)$ the SHAP method breaks down each prediction into

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i x_i = \text{bias} + \sum \text{contribution of every feature}$$

where ϕ_i is the SHAP value associated to feature x_i. The ϕ_0 is the bias term which represents the value that the model would predict in the absence of all features. The intuition of the SHAP method is that it attempts to create a model for every prediction of the original model which identifies how the presence or absence of feature x_i affected the prediction. This is done by comparing what the original model predicts when a feature is present - to the prediction when the feature is absent. However, since the order in which a model sees features is important, this comparison is done in every possible combination to create a fair comparison between features.

The SHAP values have three main desirable properties:

1. The local accuracy property which states that the sum of all of the feature effects sums to the output of the model prediction. Essentially what this means is that the sum off all the SHAP values for a particular prediction will sum to the predicted quantity.

2. The missingess property which states that if a feature is absent in the input, it should have a SHAP value of 0.

3. The consistency property which states that changing a model so a feature impacts a prediction more will never decrease the SHAP value of that feature. This property is particularly important for comparing SHAP values between different types of models.

These three properties are known to be desirable in additive feature explanation methods and the SHAP method was the first to combine all three into a unified framework.

In regards to the Whisper sensitivities dataset, a positive SHAP value shows that the model predicts an increased positive bias by the magnitude of the SHAP value when a benchmark is sensitive to a specific isotope-reaction in an energy range. Conversely, a negative SHAP value means that the model predicts an increased negative bias, when a benchmark is sensitive to that isotope-reaction in an energy range. A useful feature of SHAP values is that they are in the same units as the predicted quantity of the model. SHAP values can be thought of as sensitivity coefficients as they essentially detail what the most important features are to a machine learning model and how they affect the predictions. This is analogous to how the k_{eff} sensitivities detail what materials are present in a system and how they influence the neutron multiplication. In the context of this work, the SHAP values are essentially the sensitivity coefficients of the machine learning model to the k_{eff} sensitivities encountered in the dataset when making predictions of

the bias. In the following sections the SHAP framework is used to analyze and help interpret predictions of the machine learning model on the Whisper dataset.

III. METHODOLOGY OVERVIEW

The main goal of this methodology is to help nuclear data evaluators identify subsets of nuclear data that have the highest probability of having errors. The first step is to use the sensitivity profiles for specific groups of benchmarks (HEU-metal, ^{233}U solutions, etc.) to predict the bias using some form of machine learning algorithm. The second step is to use additive feature explanation methods (SHAP values) to identify the subsets of sensitivities that have the strongest impact on the predicted bias. Once the isotopes and reactions that have the strongest relationship with the bias are identified, a correlation analysis must be performed on the sensitivities to see what other subsets of nuclear data those isotopes and reactions are linked with. Finally these identified nuclear data subsets must be brought to a nuclear data expert to compare the evaluated values of the various libraries with differential measurements.

This approach highlights specific subsets of nuclear data which have the highest probability of causing increased errors in transport simulations. Of course an expert could also investigate nuclear data to find inconsistencies and/or errors, but the extremely large quantity of data that would have to be validated with respect to differential measurements and other nuclear data libraries makes this task almost impossible for a human to perform across all nuclear data. On the other hand, the proposed methodology performs the above task in an automated, reproducible, and unbiased manner that can quickly pick up shortcomings in nuclear data with respect to an observable. A small caveat is that the Random Forest algorithm used in this study is stochastic in nature and is only exactly reproducible when a random seed is set.

One of the reasons Random Forests are chosen to be used in this study is due to how correlations are handled in features. As mentioned earlier each tree in the Random Forest selects a random subset of features, and this results in a favorable outcome. Essentially the ensemble learns multiple relationships between the features and the predicted quantity, and as a consequence the feature importance values (as measured by SHAP) are diffused across the sensitivities. This results in a more flexible model that has multiple representations of the relationships in the dataset. As a consequence this approach tends to highlight clusters of features correlated with the target value

instead of individuals. This is useful as any individual isotope-reaction in an energy range is not likely to be the sole influencer of bias, but most likely it is part of a group of correlated reactions which are increasing errors. In general this methodology is not limited to a Random Forest and any regression algorithm can be used.

Some other algorithms that were employed as an alternative to this methodology were neural networks [23] and the XGBoost regression algorithm [24]. Both of these approaches have a strong potential to be successful in this application, but there are some differences in how the feature importance measures are calculated. The XGBoost is a regularized regression method which has weights assigned to each feature (analogously to coefficients in linear regression) [24]. The loss function to be minimized contains the mean squared error of the predicted quantity along with the L1 and L2 regularization terms which attempt to shrink the feature coefficients in order to prevent overfitting. Effectively the XGBoost algorithm does the opposite of the Random Forest in terms of feature importance measures i.e. diffusion vs. consolidation. This approach will highlight the isotope-reactions that are the most important to predicting the bias while shrinking the importance of the correlated features. It is then up to the analyst to identify all the correlations that exist within the sensitivities to identify other potential reactions which are leading to bias.

Neural networks are another class of algorithms that have a strong potential to be successful in this application. Neural networks have demonstrated state of the art performance across various physics problems where the datasets are both sparse and contain an extreme number of dimensions [25]. This is exactly the dataset we are working with where we have sensitivity profiles that contain around 90,000 values per benchmark. However, neural networks generally have a very large number of parameters to be optimized and as such perform best when the training datasets are also very large. Consequently, we were not able to match the accuracy of the Random Forest in this study, and this is most likely because our dataset contains only 1,100 benchmarks.

These comments are provided to point out that this methodology does not require the Random Forest algorithm to be used as other promising algorithms do exist. No matter what algorithm is employed, the analyst should have an strong understanding of how the algorithm works and how the choice of algorithm will impact the final feature importance measures generated by the SHAP algorithm. After thorough investigation we recommend that a 'base case' using a Random Forest is established and only after that other algorithms are utilized and the feature importance values

are investigated and compared between the two. We utilize the Random Forest as the main case as it inherently identifies complex correlations across large portions of the dataset, is able to learn non-linear relationships, and is able to learn from 'small' datasets.

Note that this methodology is not limited to criticality benchmarks or even radiation transport simulations. We merely developed this methodology within the space of applied nuclear criticality safety validation and are demonstrating the utility with respect to nuclear data. There are many other engineering disciplines that perform validation of simulations by comparing to experimental measurements. The only general requirement is that we need to be able to take the derivative of the simulation outputs with respect to the inputs, i.e. sensitivity coefficients. Given that we can generate sensitivity coefficients, we can apply this method to any field that relies on numerical simulations. In the following section we demonstrate this approach by applying this methodology to ^{233}U solution benchmarks in the Whisper benchmark library.

IV. RESULTS

IV.A. Benchmarks Used

The benchmarks used in this study are from the Whisper library with the primary focus being on the ^{233}U solution cases. The main reason for this choice of benchmarks is that the ^{233}U solutions are known to have large bias in criticality calculations.

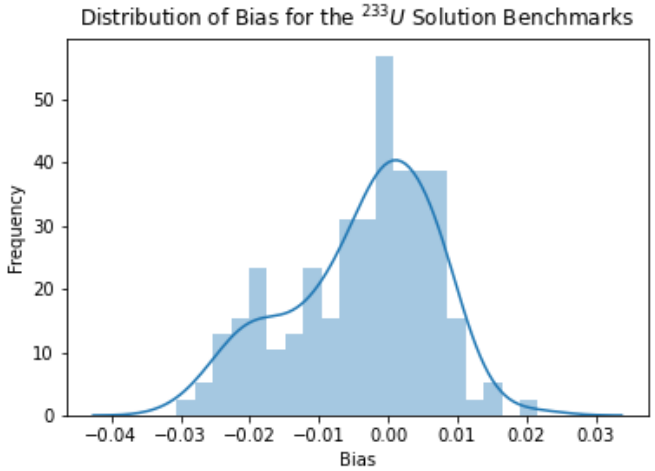

Fig. 1. The distribution of bias for the ^{233}U solution benchmarks.

The bias distribution for these cases is summarized in Figure 1. The distribution is skewed toward the negative direction which means that the simulated k_{eff} is often below the experimentally measured k_{eff}. Past studies have investigated adjustment to the ^{233}U nuclear data to minimize the bias for these benchmarks, and the conclusion was that the nuclear data needs to be revisited as it is one of the primary causes of increased bias [26]. For this reason we are focusing on the ^{233}U solution benchmarks to demonstrate how we can identify specific subsets of nuclear data which are likely causing increased bias in the calculation. However, we are not focusing specifically on the ^{233}U nuclear data, but instead investigating all isotopes which the benchmarks are sensitive to.

IV.B. Predicting Bias

The first step in this process is to train a machine learning algorithm to learn the relationships between the bias and the sensitivity profiles for these benchmarks. Then we can apply SHAP values to the machine learning model to identify the complex relationships with the goal of identifying subsets of nuclear data that have the highest probability of influencing bias. The main assumption in this step is that sensitivities are able to fully characterize a system, and as a result can be used to find patterns in the sensitivities that are related with the bias. The main challenge with this approach is overcoming the correlations that exist within the sensitivities among neighboring energy bins, between reactions of a single isotope, and reactions of different isotopes. If two sensitivities are strongly correlated then the Random Forest can use them interchangeably during the training process. This diffuses the feature importance between correlated sensitivities, and as a result we must perform a correlation analysis and then combine that with expert judgement to identify problematic nuclear data.

To reduce the diffusion across nearby energy groups the energy grid was converted from a 44 group to a 11 group by summing every 4 sensitivities for each reaction. The resulting group structure can be found in the Appendix. A Random Forest model was trained on the 11-group sensitivity profiles for the benchmarks containing ^{233}U solutions. The model performance using all isotope sensitivities compared with only using ^{19}F sensitivities is summarized in Table I.

Table I illustrates that the majority of the bias can be predicted using just ^{19}F sensitivities. The coefficient of determination (R^2), which details the proportion of the variance in the bias that

Data	Mean Absolute Error	R^2
Full Sensitivity Profile	0.00384	0.745
^{19}F	0.00429	0.732

TABLE I
The resulting model performance when the full sensitivity profile is used compared to when just ^{19}F sensitivities are used. Obtained from 20 fold cross validation.

is explained by the inputs, goes down by about 1.74% and the mean absolute error increases by 10% when the features are limited to ^{19}F sensitivities. Later on in the paper we demonstrate that ^{19}F nuclear data is problematic using SHAP analysis, and the strong performance of the model using only this isotope is the first indication of that. For the ^{233}U benchmarks used in this study the average experimental uncertainty in the k_{eff} measurements is 0.00487 and from Table I we can see that the mean absolute error for both datasets is below that. The fact that the model performs relatively well with just ^{19}F indicates that this isotope has a strong relationship with the bias. Intuitively, this leaves two primary drivers of bias related to nuclear data in the ^{233}U solution systems: 1) ^{19}F nuclear data, or 2) something highly correlated with ^{19}F nuclear data. Note that if either of these reasons do not appear to be the root cause of the bias, then further investigation into the validity of ^{233}U solution benchmarks must be considered. To explore this further we perform a SHAP analysis for both of these datasets.

IV.C. Explaining Bias Prediction

The SHAP values can be thought of as sensitivity coefficients of the machine learning model predictions with respect to the k_{eff} sensitivities encountered in the dataset. We can use a SHAP summary plot (see Figure 2) to identify what sensitivities are important to predicting the bias for the ^{233}U solution benchmarks when trained using the full sensitivity profile.

In Fig. 2 the color corresponds to the sensitivity value and the position on the x-axis corresponds to the impact on the bias prediction. Every point in a row details how a prediction for a benchmark was affected by the sensitivity value for that reaction. The distribution of SHAP values for each isotope-reaction represents how the predicted bias is shifted away from the mean prediction over the range of sensitivities encountered in the dataset. The reason for the roughly symmetric pattern around zero is that the model has to make predictions for benchmarks that are sensitive to a specific reaction, and also for other benchmarks that are not sensitive to that specific

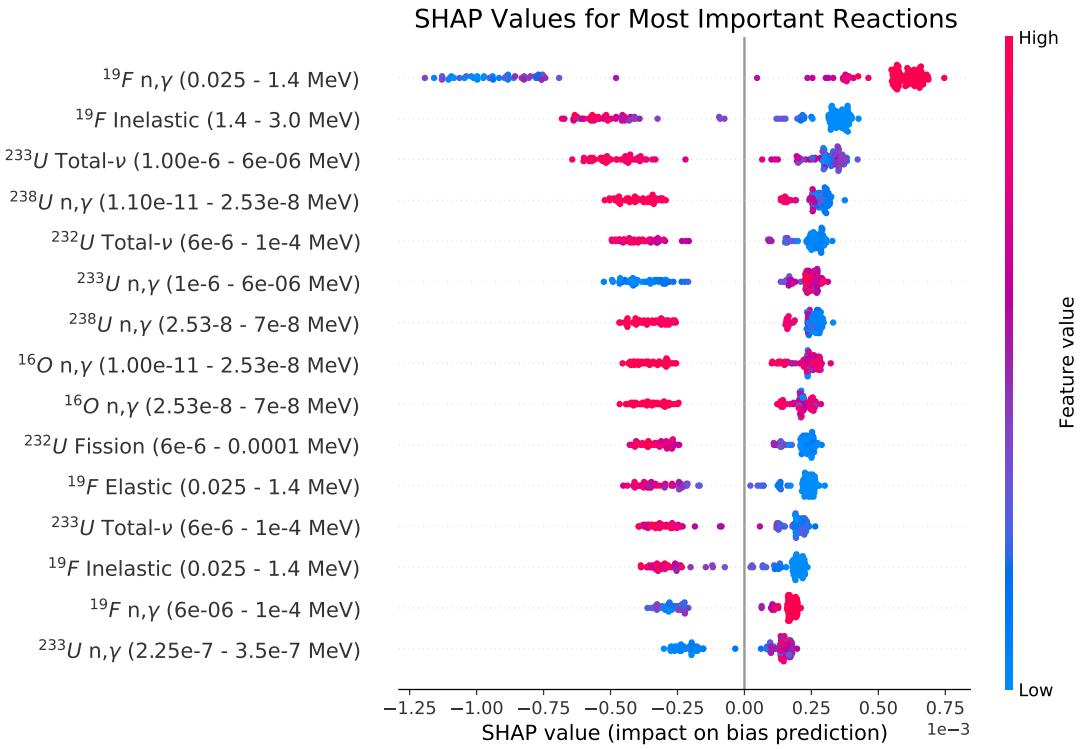

Fig. 2. The SHAP values of the top 15 most important features on every benchmark (each point is a sample), when trained on the ^{233}U solution cases using the full sensitivity profile. The plot sorts features by the sum of SHAP value magnitudes over all samples, and uses the SHAP values to show the distribution of the impacts of each feature on the model output.

reaction. When a benchmark is sensitive to a bias-inducing reaction, the model adds in the effect on the prediction of bias, and when the reaction is not present, the model has to remove the effect.

From Figure 2 we can see that ^{19}F takes up the top two spots and accounts for 5 out of the 15 top reactions ranked by SHAP values. As a result we are mainly focusing on ^{19}F nuclear data in this research. We can see that the ^{19}F capture reaction has the strongest effect on the predicted bias, and that when the k_{eff} sensitivity for that reaction is strong (blue) the bias prediction is pulled in the negative direction (negative SHAP value). However, a negative SHAP value does not necessarily mean the bias is reduced, it just means that the predicted bias is pushed in the negative direction, which corresponds to a lower simulated k_{eff}. This relationship can be further explored using a partial-dependence plot shown in Figure 3.

The ^{19}F(n,γ) reaction has a negative sensitivity coefficient in this energy range, and when benchmarks are sensitive to it, the model predicts a larger negative bias (negative SHAP value)

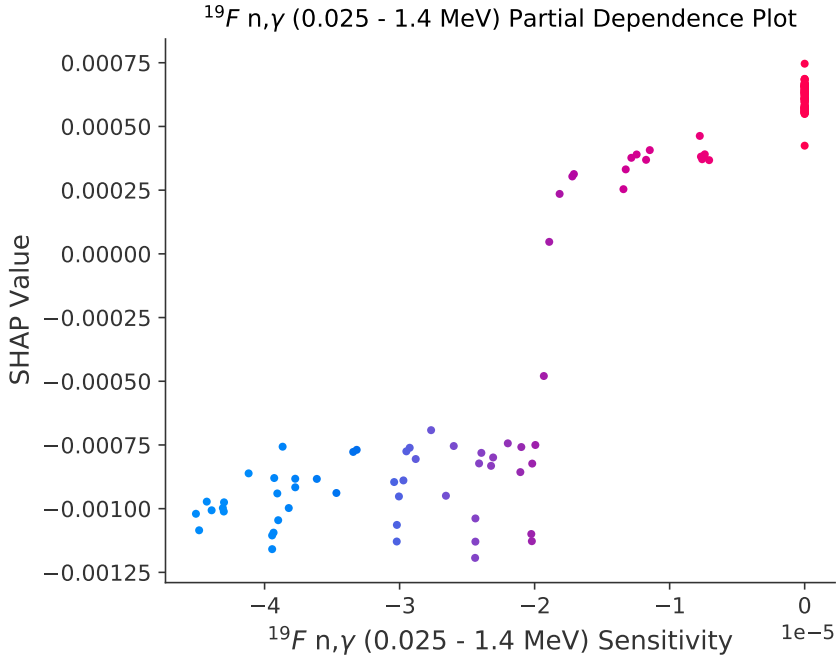

Fig. 3. Each point in the figure is a sensitivity encountered in the dataset with the sensitivity value shown on the x-axis and the SHAP value on the y-axis. It can be seen that when the sensitivity for this reaction reaches a critical value (approximately -2e-5) the model begins to predict a lower bias (corresponds to a decreased simulated k_{eff}).

which corresponds to a decreased simulated k_{eff}. From this we can infer that the machine learning model is learning that this reaction is pushing the simulated k_{eff} too far in the negative direction. Essentially the model is learning that the reaction is introducing too much negative reactivity across multiple benchmarks which is causing the simulated k_{eff} to be depressed in the MCNP6 calculations. A possible explanation for this result is that the ^{19}F(n,γ) cross section might be too large in this energy range, which is leading to a depressed simulated k_{eff} (larger negative bias). In general it can be postulated that a cross section is too large if the sensitivity and the SHAP value are the same sign, and conversely if they have opposite signs the cross section can be postulated to be too small.

A quick way to identify the inference on the cross section magnitude is to identify the slope on the SHAP vs. sensitivity plot. If we were to plot a line of best fit through Fig. 3 we can see that the slope on the data is positive, and in this case since the SHAP values and sensitivity are the same sign the cross section is inferred to be too large. If we extend the idea to the opposite quadrant (positive sensitivity and positive SHAP) the slope is yet again positive and we again

have too large of a cross section. Thus we can directly relate the slope of the SHAP vs. sensitivity data to the relative magnitude of the cross section error. If the slope is positive then the cross section is too large and if the slope is negative the cross section is too small (according to the ML algorithm). This relationship is summarized in Figure 4.

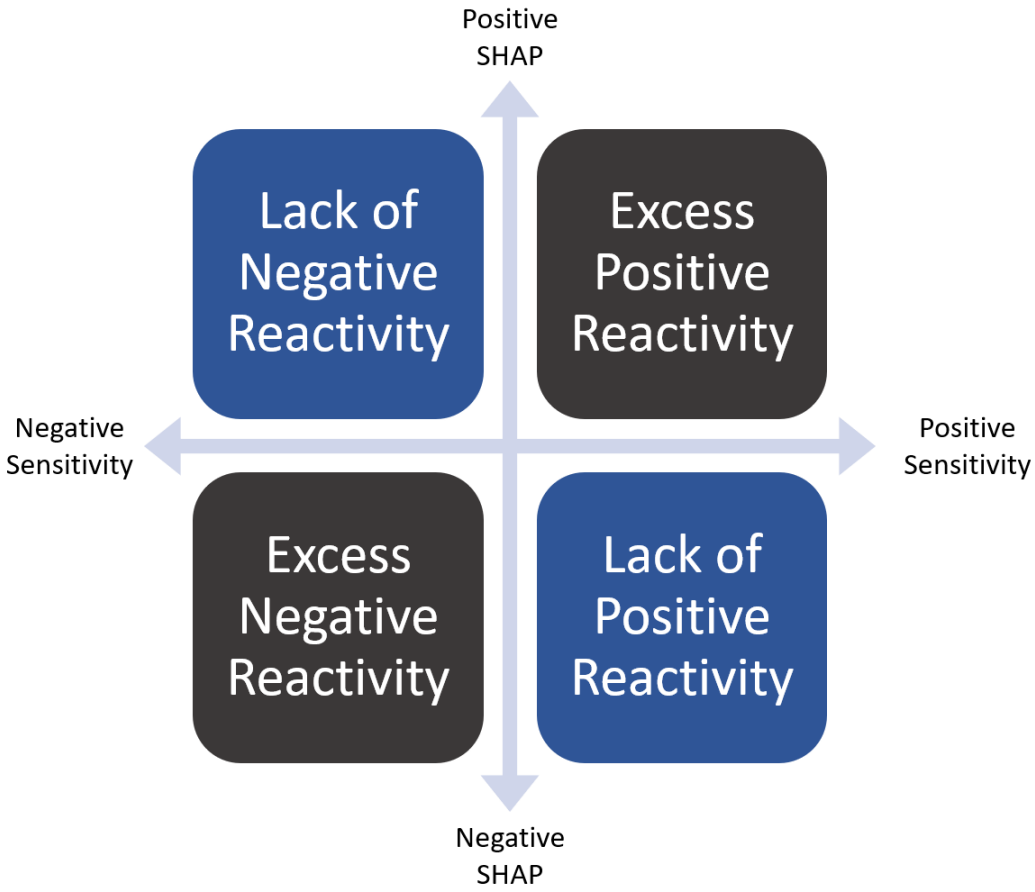

Fig. 4. A summary of the insights obtained from the SHAP vs. sensitivity analysis. If a sensitivity-SHAP point lands in a grey square the nuclear data is inferred to be too large. and if it lands in a blue square the nuclear data is inferred to be too small.

This is very useful if one wishes to analyze large quantities of data and identify what the machine leaning insights are in regions of nuclear data.

The main goal of this section was to identify some candidate isotopes that could potentially have problematic nuclear data. From Figure 2 we can see that there is a mix of actinides and minor elements like ^{16}O, but the most common isotope is ^{19}F, and for that reason we will focus on it.

IV.C.1. Identifying Problematic Nuclear Data

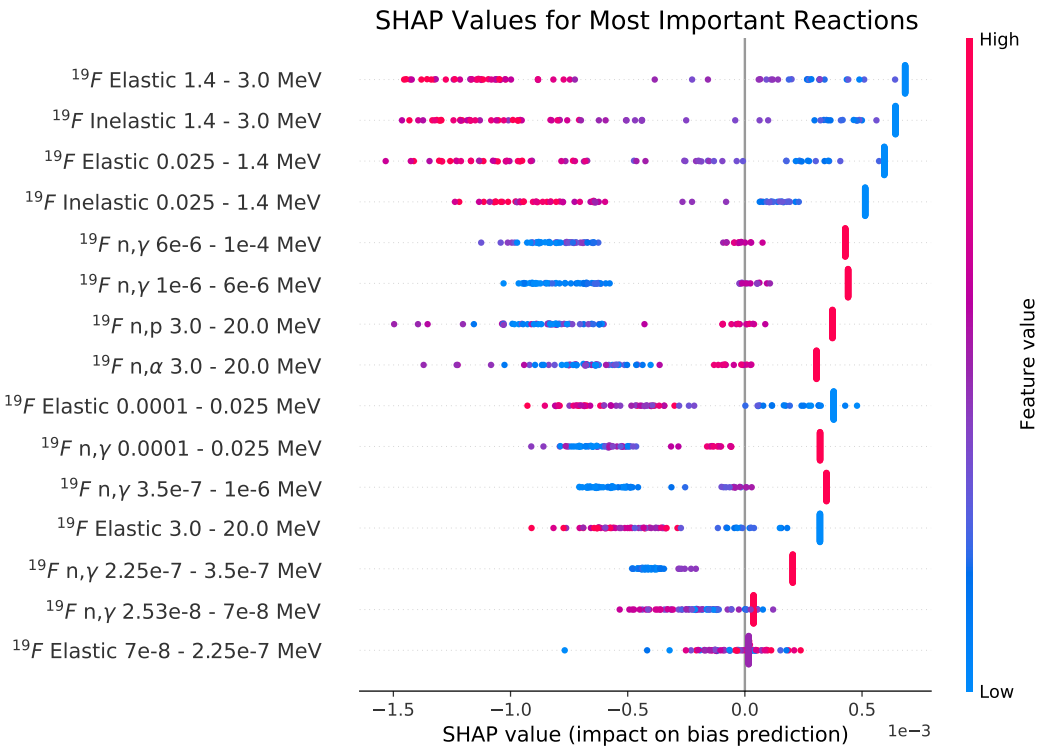

Fig. 5. The SHAP values of the top 15 most important features on every benchmark (each point is a sample), when trained on the ^{233}U solution cases using only the ^{19}F sensitivity profile. The plot sorts features by the sum of SHAP value magnitudes over all samples, and uses the SHAP values to show the distribution of the impacts of each feature on the model output.

To investigate the relationships between ^{19}F and the bias we trained a Random Forest on just the ^{19}F sensitivity profiles and the SHAP summary plot can be found in Figure 5. The top most important reaction is elastic scattering followed by inelastic scattering between 1.3 and 3.0 MeV. The next two important reactions are the elastic/inelastic in the adjacent lower energy bin for the same reaction. It is of interest to identify the correlations that exist with these most important reactions. As was stated previously, if groups of sensitivities are strongly correlated, then the Random Forest can use those sensitivities interchangeably during the prediction process, and as a result the SHAP importance is diffused. The top 10 most correlated sensitivities with the ^{19}F Elastic 7e-8 - 2.25e-7 MeV reaction are shown in Table II.

In Table II there are numerous reactions with very strong correlations with the feature of interest, and it is not immediately clear which of these correlated sensitivities are indeed influencing

Reaction	Energy Range	Correlation Coefficient
^{19}F Inelastic	0.025 - 1.4 MeV	0.996
^{19}F Inelastic	1.4 - 3.0 MeV	0.996
^{19}F Elastic	0.025 - 1.4 MeV	0.994
^{19}F Inelastic	3.0 - 20.0 MeV	0.993
^{19}F Elastic	3.0 - 20.0 MeV	0.982
...	...	...
^{19}F n,γ	1.4 - 3.0 MeV	-0.992
^{19}F n,α	1.4 - 3.0 MeV	-0.992
^{19}F n,γ	0.025 - 1.4 MeV	-0.992
^{19}F n,γ	0.0001 - 0.025 MeV	-0.993
^{19}F n,γ	6e-6 - 0.0001 MeV	-0.993

TABLE II
The top 10 most correlated sensitivities with the ^{19}F Elastic 7e-8 - 2.25e-7 MeV sensitivity.

the bias. However this correlation list narrows down (and ranks) the potential isotope reactions that one would have to search through to find the true influencer(s) of bias. This is the point at which a nuclear data expert must evaluate these selected isotope reactions across these energy bins. These results regarding ^{19}F were presented to nuclear data experts at Los Alamos National Laboratory (LANL) and they investigated the ^{19}F nuclear data. They compared various nuclear data libraries along with differential measurements and concluded that the ^{19}F inelastic nuclear data is indeed erroneous [8]. A brief overview of the expert-based analysis of the ^{19}F inelastic cross section deficiencies is provided next.

IV.C.2. Investigating Identified Nuclear Data

The first step of investigating the quality of nuclear data is to compare the various libraries and differential measurements. For example, the ^{19}F inelastic reaction in the energy range identified as important by the ML algorithm is shown in Figure 6.

Figure 6, generated through the online interface of the National Nuclear Data Center (NNDC) [27] evaluated and experimentally measured nuclear data databases, shows that the JENDL-4.0 [28] nuclear data has a stronger agreement with differential measurements than ENDF/B-VIII.0 [29] between 0.3 - 1.0 MeV. This difference is mainly due to the ENDF/B evaluation extending the resonance region higher in incident neutron energy than JENDL. Additionally if we perform the SHAP slope analysis on this reaction we find a positive sensitivity and negative SHAP which implies the cross section is too small. The divergence of the ENDF/B nuclear data from the differential measurements and JENDL is a clear lapse in the evaluation and it was correctly identified by the

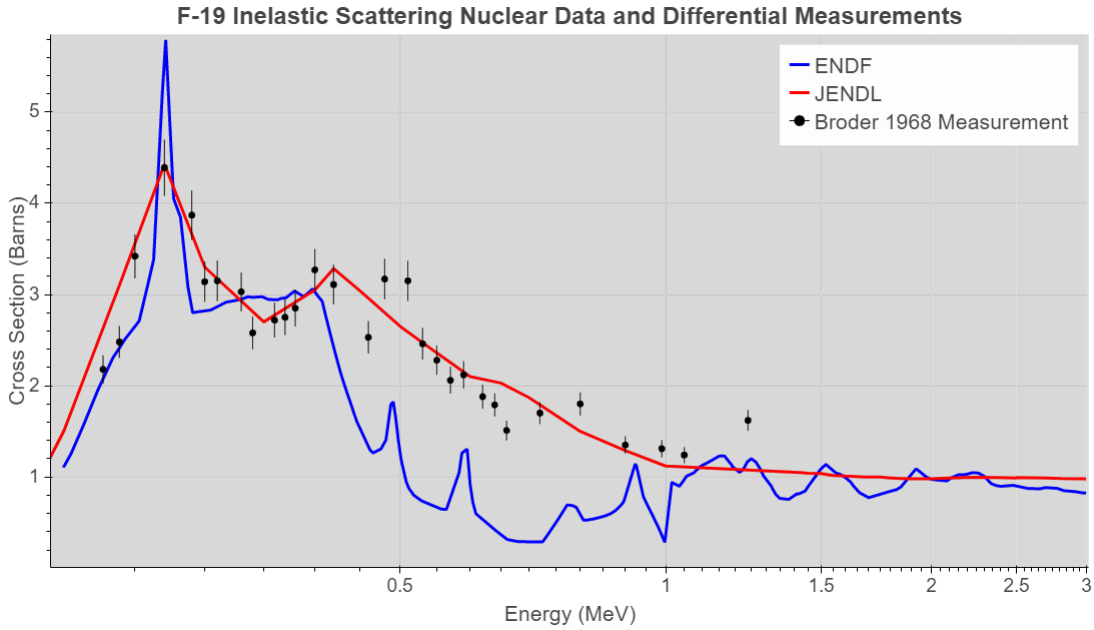

Fig. 6. A comparison of ENDF/B-VIII.0 and JENDL-4.0 nuclear data for ^{19}F along with the differential measurements in the region identified as important by the ML algorithm.

ML algorithm. This result demonstrates the core of this methodology: using ML to identify regions in nuclear data that have the highest probability of needing attention through a reevaluation of the cross sections. A nuclear data expert would have no problem identifying the problematic ENDF/B evaluation given in Figure 6, but they would first need to have a reason to look there. Most likely the ^{19}F nuclear data has been overlooked as it is most likely deemed to have a negligible impact to the bias by experts.

This is exactly where the strengths of the ML approach come into fruition. A nuclear data expert hoping to find erroneous data would have to go through a large number of nuclear data files, while considering both experimental measurements and integral benchmarks. This approach would be biased as the expert would most likely search through isotopes and reactions that they think are important, while neglecting trace isotopes which could be hiding errors. Additionally, the large quantity of nuclear data available makes this quite a monumental task for a team, let alone an individual.

On the other hand, the ML algorithm is able to process hundreds of benchmarks in order to learn how small variations in the sensitivity profiles across all isotopes are related to small variations in the bias. We can then use SHAP values to investigate exactly how the ML algorithm

is using the sensitivity values to make predictions, which provides insights about relationships present between nuclear data and the bias. We have shown here that this approach is able to pick up faults in nuclear data in an automatable, reproducible, and unbiased manner. We do want to stress that the role of the nuclear data expert is extremely important. The ML tool is not meant to replace expert judgement, but instead be a tool to help nuclear data experts identify problematic nuclear data subsets.

V. CONCLUSION

As the quantity and quality of nuclear experiments continues to increase, machine learning algorithms provide an avenue to help experts make sense of the large amounts of data. Machine learning algorithms have been shown to be able to accurately predict the bias of MCNP6 criticality simulations, and that we can use SHAP values to extract relationships between the sensitivities and the bias. The insights can point us at the nuclear data which has the highest probability of indeed causing increased errors in simulations. These problematic nuclear data subsets can then be "fixed" by applying nuclear data assimilation methods, or by reevaluating the problematic nuclear data with updated physics models or new experimental data.

In this manuscript we have proposed a novel methodology to help identify problematic subsets of nuclear data. The benefits of having more accurate nuclear data include improved safety, more efficient nuclear reactor operation, and enhanced production of radioisotopes. This approach is not limited to criticality benchmarks, but can be applied to shielding, depletion, fixed-source, and high-energy physics benchmarks. In general the requirements of this methodology are that we have experimental measurements, we have simulations of those experiments, and that we are able to compute sensitivities of the simulation outputs with respect to the inputs. If those three criteria are met, this methodology could be applied to many other disciplines that employ numerical simulation.

ACKNOWLEDGMENTS

The authors would like to thank D. Neudecker for her valuable review and feedback of this manuscript. This work was partly supported by the DOE Nuclear Criticality Safety Program, and partly supported by the Advanced Scientific Computing Program, Physics and Engineering Nuclear Physics subprogram, funded and managed by the National Nuclear Security Administration (NNSA). Work at LANL was carried out under the auspices of the NNSA of the U.S. Department of Energy under contract 89233218CNA000001.

APPENDIX

Group Number	Lower Energy (MeV)	Upper Energy (MeV)
0	1.01e-11	2.53e-08
1	2.53e-08	7.00e-08
2	7.00e-08	2.25e-07
3	2.25e-07	3.50e-07
4	3.50e-07	1.00e-06
5	1.00e-06	6.00e-06
6	6.00e-06	1.00e-04
7	1.00e-04	0.025
8	0.025	1.4
9	1.4	3.0
10	3.0	20.0

TABLE III
Energy group structure used in this work which was converted from the 44 group originally used in Whisper. All units are in MeV.

REFERENCES

[1] L. Bernstein, D. Brown, A. Hurst, J. Kelly, F. Kondev, E. McCutchan, C. Nesaraja, R. Slaybaugh, and A. Sonzogni, "Nuclear data needs and capabilities for applications," *arXiv preprint arXiv:1511.07772* (2015).

[2] M. Chadwick, M. Herman, P. Obložinskỳ, M. E. Dunn, Y. Danon, A. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla et al., "ENDF/B-VII. 1 nuclear data for science and technology: cross sections, covariances, fission product yields and decay data," *Nuclear data sheets*, **112**, *12*, 2887 (2011).

[3] B. L. Broadhead, B. T. Rearden et al., "Sensitivity-and uncertainty-based criticality safety validation techniques," *Nuclear science and engineering*, **146**, *3*, 340 (2004).

[4] C. Werner, J. Bull, C. Solomon et al., "MCNP6.2 Release Notes," LA-UR-18-20808, Los Alamos National Laboratory (2018).

[5] P. Grechanuk, M. Rising, and T. Palmer, "Using Machine Learning Methods to Predict Bias in Nuclear Criticality Safety," *Journal of Computational and Theoretical Transport* (in press).

[6] B. Kiedrowski, F. Brown, J. Conlin et al., "Whisper: Sensitivity/uncertainty-based computational methods and software for determining baseline upper subcritical limits," *Nuclear Science and Engineering*, **181**, *1*, 17 (2015).

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, 4765–4774 (2017).

[8] D. Neudecker, M. Grosskopf, M. Herman, W. Haeck, P. Grechanuk, S. Vander Wiel, M. Rising, A. Kahler, N. Sly, and P. Talou, "Enhancing nuclear data validation analysis by using machine learning," *Nuclear Data Sheets*, **167**, 36 (2020).

[9] F. B. Brown et al., "User Manual for Whisper-1.1," LA-UR-17-20567, Los Alamos National Laboratory (2017).

[10] B. C. Kiedrowski, F. B. Brown et al., "Whisper: Sensitivity/uncertainty-based computational methods and software for determining baseline upper subcritical limits," *Nuclear Science and Engineering*, **181**, *1*, 17 (2015).

[11] J. T. Goorley, M. James, T. Booth, F. Brown et al., "Initial MCNP6 Release Overview," *Nuclear Technology*, **180**, *3* (2012).

[12] B. T. Rearden, "Perturbation theory eigenvalue sensitivity analysis with Monte Carlo techniques," *Nuclear science and engineering*, **146**, *3*, 367 (2004).

[13] B. C. Kiedrowski, "Review of Early 21st-Century Monte Carlo Perturbation and Sensitivity Techniques for k-Eigenvalue Radiation Transport Calculations," *Nuclear Science and Engineering*, **185**, *3*, 426 (2017); 10.1080/00295639.2017.1283153., URL https://doi.org/10.1080/00295639.2017.1283153.

[14] F. B. Brown, M. E. Rising et al., "MCNP-WHISPER Methodology for Nuclear Criticality Safety Validation," LA-UR-16-23757, Los Alamos National Laboratory (2016).

[15] J. B. Briggs, L. Scott, and A. Nouri, "The international criticality safety benchmark evaluation project," *Nuclear science and engineering*, **145**, *1*, 1 (2003).

[16] B. Kiedrowski and F. Brown, "Methodology, verification, and performance of the continuous-energy nuclear data sensitivity capability in MCNP6," *Proceedings of the 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering* (2013).

[17] C. M. Bishop, *Pattern recognition and machine learning*, springer (2006).

[18] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," *Proceedings of the twenty-first international conference on Machine learning*, 114, ACM (2004).

[19] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, **12**, *Oct*, 2825 (2011).

[20] D. G. Denison et al., "A bayesian CART algorithm," *Biometrika*, **85**, *2*, 363 (1998).

[21] A. Liaw et al., "Classification and regression by randomForest," *R news*, **2**, *3*, 18 (2002).

[22] L. Breiman, "Random forests," *Machine learning*, **45**, *1*, 5 (2001).

[23] M. A. Nielsen, *Neural networks and deep learning*, vol. 2018, Determination press San Francisco, CA (2015).

[24] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794 (2016).

[25] X. Ju, S. Farrell, P. Calafiura, D. Murnane, L. Gray, T. Klijnsma, K. Pedro, G. Cerati, J. Kowalkowski, G. Perdue et al., "Graph neural networks for particle reconstruction in high energy physics detectors," *arXiv preprint arXiv:2003.11603* (2020).

[26] M. E. Rising, F. B. Brown, and J. L. Alwin, "Using Whisper-1.1 to Guide Improvements to Nuclear Data Evaluations," , Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2017).

[27] "Evaluated Nuclear Data File (ENDF)," *NRDC-Network* (2018)URL www.nndc.bnl.gov/exfor/endf00.

[28] K. Shibata, O. Iwamoto, T. Nakagawa, N. Iwamoto, A. Ichihara, S. Kunieda, S. Chiba, K. Furutaka, N. Otuka, T. Ohsawa, T. Murata, H. Matsunobu, A. Zukeran, S. Kamada, and J. Katakura, "JENDL-4.0: A New Library for Nuclear Science and Engineering," *J. Nucl. Sci. Technol.*, **48(1)**, 1 (2011).

[29] D. Brown, M. Chadwick, R. Capote, A. Kahler, A. Trkov, M. Herman, A. Sonzogni, Y. Danon, A. Carlson, M. Dunn, D. Smith, G. Hale, G. Arbanas, R. Arcilla, C. Bates, B. Beck, B. Becker, F. Brown, R. Casperson, J. Conlin, D. Cullen, M.-A. Descalle, R. Firestone, T. Gaines, K. Guber, A. Hawari, J. Holmes, T. Johnson, T. Kawano, B. Kiedrowski, A. Koning, S. Kopecky, L. Leal, J. Lestone, C. Lubitz, J. M. Damián, C. Mattoon, E. McCutchan, S. Mughabghab, P. Navratil, D. Neudecker, G. Nobre, G. Noguere, M. Paris, M. Pigni, A. Plompen, B. Pritychenko, V. Pronyaev, D. Roubtsov, D. Rochman, P. Romano, P. Schillebeeckx, S. Simakov, M. Sin, I. Sirakov, B. Sleaford, V. Sobes, E. Soukhovitskii, I. Stetcu, P. Talou, I. Thompson, S. van der Marck, L. Welser-Sherrill, D. Wiarda,

M. White, J. Wormald, R. Wright, M. Zerkle, G. Žerovnik, and Y. Zhu, "ENDF/B-VIII.0: The 8th Major Release of the Nuclear Reaction Data Library with CIELO-project Cross Sections, New Standards and Thermal Scattering Data," *Nuclear Data Sheets*, **148**, 1 (2018); https://doi.org/10.1016/j.nds.2018.02.001., URL http://www.sciencedirect.com/science/article/pii/S0090375218300206, special Issue on Nuclear Reaction Data.