SAND2022-8249C

2022 International High-Level Radioactive Waste Management Conference

Machine Learning Surrogate Process Models for Efficient Performance Assessment of a Nuclear Waste Repository

Bert J. Debusschere*, D. Thomas Seidl**, Timothy M. Berg*, Kyung Won Chang**, Rosemary C. Leone**, Laura P. Swiler**, and Paul E. Mariner**

*Sandia National Laboratories, P.O. Box 969, Livermore, CA 94551-0969 USA, bjdebus@sandia.gov
** Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87123-0747 USA

Performance assessment of a nuclear waste repository requires tracking the evolution of thousands of waste packages over time. While detailed process models for these packages, such as the Fuel Matrix Degradation model (FMD), are available, they are too computationally expensive to be used in a comprehensive full-repository simulation, especially if multiple samples of the repository simulation are desired for uncertainty quantification. In this work, we evaluate the accuracy of two Machine Learning (ML) methods: a k-Nearest Neighbor regressor (k-NNr) and an Artificial Neural Network (ANN), to be used as computationally efficient approximations to the FMD process model. Preliminary results show that these ML surrogates can be very accurate, especially if some information about the internal fuel cask state is included as a feature in the training data.

I. INTRODUCTION

The Geologic Disposal Safety Assessment (GDSA) *Framework* is open source repository simulation software built around the massively-parallel multi-physics code PFLOTRAN[1]. An important short-term goal of the development of the *GDSA Framework* (pa.sandia.gov) is to perform probabilistic repository simulations to identify sources of uncertainty to help prioritize future R&D. To achieve this short-term goal with current computer resources, developers must consider ways to include the effects of expensive process models in total system simulations.

High fidelity prediction of waste package and waste form degradation processes for thousands of waste packages in a probabilistic repository performance assessment calculation is expensive. With thousands of waste packages, thousands of time steps, and hundreds of realizations in a simulation to allow for uncertainty quantification, these process models may need to be called a billion times per simulation.

One way to reduce computational expense is to develop response surface surrogate models that can rapidly emulate the mechanistic process models. An ideal response surface surrogate model runs orders of magnitude faster than its parent mechanistic model and provides outputs identical to those of the mechanistic model within a specified range of the model inputs.

Over the past few years, a team of modelers and mathematicians at Sandia National Laboratories has been developing surrogate models for the UO_2 Flux that is predicted by the Fuel Matrix Degradation (FMD) process model[2]. The FMD model has been coupled with PFLOTRAN[3], but the coupled model runs too slowly for a set of probabilistic repository-scale simulations. The surrogate modeling work has examined Machine Learning (ML) approaches such as tabulation with tree-based lookup methods, and artificial neural networks. A key question for obtaining good accuracy with the surrogate models is the choice of features to train the surrogates on, as well as the sampling approaches used to generate training data. Section II in this paper describes the FMD process model, and Section III discusses our general approach for surrogate modeling along with the process for generating training data. Section IV then shows some results from surrogates that do not use detailed information about the internal state of the waste packages and waste form. Section V explores the use of surrogate models that incorporate features internal to the fuel cask, along with preliminary results.

II. FUEL DISSOLUTION PROCESS MODEL

The FMD model is a mechanistic spent fuel dissolution model coded in Matlab and developed at Argonne National Laboratory and Pacific Northwest National Laboratory. The model calculates spent fuel dissolution rates as a function of radiolysis, alteration layer growth, diffusion of reactants through the alteration layer, temperature, and interfacial corrosion potential[4]. It employs a one-dimensional (1D) reactive transport model to simulate diffusion and chemical reactions across this layer over time. The 1D domain, depicted in Fig. 1, extends 0.05 m from the fuel surface to the bulk water. It is divided into as many as 100 cells with increasing length toward the bulk water boundary cell.

To couple the FMD model with PFLOTRAN, a "coupled" FMD model was coded in Fortran. At each time step, PFLOTRAN calls the coupled FMD model to obtain

a new dissolution rate. Coupling required PFLOTRAN to keep track of the 1D chemical profiles across the domain from the previous time step. It also required relevant inputs from the main PFLOTRAN simulation, such as temperature, time, and environmental concentrations in the boundary cell. Dose rate is calculated in the coupled FMD model from time and burnup. A full list of FMD model inputs and outputs available for surrogate modeling is presented in Table I.

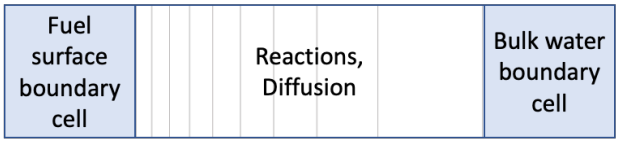

Fig. 1. FMD model domain.

TABLE I. Inputs/Outputs of Coupled FMD Model

Available Inputs	Outputs
• Initial concentration profiles across 1D corrosion/water layer (UO_2(s), UO_3(s), UO_4(s), H_2O_2, UO_2^{2+}, UCO_3^{2-}, UO_2, CO_3^{2-}, O_2, Fe^{2+}, and H_2) • Initial corrosion layer thickness • Dose rate at fuel surface (= f (time, burnup)) • Temperature • Time and time step length • Environmental concentrations (CO_3^{2-}, O_2, Fe^{2+}, and H_2)	• Final concentration profiles across 1D corrosion/water layer • Final corrosion layer thickness • Fuel dissolution rate (UO_2 flux)

The coupled Fortran FMD model was tested on a problem involving a two-dimensional flow field containing 4 rows of 13 breached spent fuel waste packages. The model successfully simulated fuel dissolution for each of the waste packages over 100 time steps[3]. Of the 45 minutes of computational time required to run the simulation, 30 minutes were used calculating the fuel dissolution rates in the coupled FMD model.

III. SURROGATE MODELING

It is often useful to construct a surrogate model to use in uncertainty and sensitivity analysis of a computational physics model when it is computationally demanding. A surrogate model (sometimes called meta-model, emulator, or response surface model) is an inexpensive input-to-output mapping that replaces a simulation code. Once constructed, this meta-model is relatively inexpensive to evaluate so it is often used as a surrogate for the physics model in uncertainty propagation, sensitivity analysis, or optimization problems that may require thousands to millions of function evaluations[5].

There are many different types of surrogate models, including neural networks [6,7], k-Nearest Neighbor regression[7], and polynomial chaos expansions[8,9]. Another popular approach in the literature is to develop an emulator that is a stationary smooth Gaussian process[10,11]. The popularity of Gaussian processes is due to their ability to model complicated functional forms and to provide an uncertainty estimate of their predicted response value at a new input point. There are many good overview articles that compare various meta-model strategies. Various smoothing predictors and nonparametric regression approaches are compared elsewhere[5,11,12]. Simpson et al. provides an excellent overview not just of various statistical meta-model methods but also approaches that use low-fidelity models as surrogates for high-fidelity models[5]. Haftka and his students developed an approach that uses ensembles of emulators or hybrid emulators[13].

Two ML surrogate modeling approaches are used in this work to predict the UO_2 flux resulting from fuel degradation: A k-nearest-neighbors surrogate model (Section III.B) and an Artificial Neural Network (Section III.C). The former interpolates between points in a high-dimensional lookup table generated by sampling the FMD model. The latter fits a nonlinear functional representation to the FMD model data. Both approaches require a sufficient amount of training data from the FMD model, as discussed in Section III.A.

III.A. Training Data

We used a standalone MATLAB implementation of the FMD process model to generate training data by randomly sampling the inputs to the model. The training data itself can be very large. For example, we may have millions of samples of FMD, where each sample involves a multi-dimensional vector sample of inputs such as the environmental concentrations, temperature, burnup, etc (the left-hand quantities in Table I). The output is also extensive, since each FMD run involves a hundred timesteps with lots of information about the fuel cask state reported at every time step (e.g., the right-hand quantities in Table I). Note that in this work, we focus on predicting the fuel dissolution rate (UO_2 flux), although the other two output quantities could be treated with a surrogate in similar manner.

A Latin hypercube sampling (LHS) study was performed to generate training and validation data for regression from the standalone FMD process model. LHS is a stratified sampling technique that generates "well-spaced" samples; it typically gives lower variance statistical estimators than plain Monte Carlo sampling[14]. The six-dimensional sample space contained the parameters initial temperature, burnup, and the

environmental concentrations of CO_3^{2-}, O_2, Fe^{2+}, and H_2. The probability distributions for each parameter are given in Table II.

TABLE II. Input Parameters and Their Distributions

Parameter	Distribution	Min.	Max.
Init. Temp. (K)	Uniform	300	400
Burnup (Gwd/MTU)	Uniform	40	65
Env. CO_3^{2-} (mol/m^3)	Log-uniform	10^{-3}	$2x10^{-2}$
Env. O_2 (mol/m^3)	Log-uniform	10^{-7}	10^{-5}
Env. Fe^{2+} (mol/m^3)	Log-uniform	10^{-3}	10^{-2}
Env. H_2 (mol/m^3)	Log-uniform	10^{-5}	$2x10^{-2}$

The temporal discretization in each problem consists of 101 logarithmically-spaced (base 10) points from 0 to 10^5 years. Some FMD runs need to be filtered out if they either get stuck in an infinite loops and never finish or if they show unphysical results, such as the UO_2 surface flux stagnating after 10^4 years, or the Corrosion Layer Thickness (CLT) growing beyond the computational domain of 0.05m.

To assess the accuracy of the models for a specific training data size, we analyzed the normalized root mean squared error (nrmse), which is computed over the data set as:

$$nrmse = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{pred,i}-y_{true,i}\right)^2}}{\frac{1}{N}\sum_{i=1}^{N} y_{true,i}} \quad (1)$$

Where N is the total number of data points. In other words, the nrmse is the root mean squared error normalized by the mean of the true data. Another metric is the mean absolute percentage error (mape), which is computed as:

$$mape = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_{pred,i}-y_{true,i}}{y_{true,i}}\right| \times 100 \quad (2)$$

The mape error, due to its relative nature, does a good job of treating the approximations in all quantities, large or small, with equal importance. On the other hand, the mape can be sensitive to numerical noise, for example when reasonable errors get divided by very small quantities in absolute value. Also, for some applications, the approximation of the larger values is the most important criterion. For these situations, the nrmse is a good overall measure of goodness. For a data set where the Quantity of Interest (QoI) spans many orders of magnitude, it is good to consider both metrics.

III.B. k-Nearest Neighbor Regression

The k-Nearest Neighbors regressor (kNNr)[7] is a supervised, non-parametric machine learning method that tabulates data points inside of a domain X with labels Y. The label for a point within the domain but not in the "table" is obtained as a weighted average of the labels of the k nearest neighbors of this new point, where $k \geq 1$ is fixed. The definition of nearest depends on the metric function one uses, though a typical choice is the Minkowski metric $\left(\sum_{i=1}^{d}|x_i - y_i|^p\right)^{\frac{1}{p}}$, with $p \geq 1$. The case of $p = 2$ is the popular Euclidean metric, which is used in this work. For efficient look-up in high-dimensional data sets, a K-D Tree tabulation method is used[6]. The inverse distances from the nearest-neighbor table points to the query point are used as the weight in the interpolation, so that points further away from the query point have less influence than more nearby points.

One of the attractive features of kNNr is that it makes predictions based on local information only, and therefore does not require global smoothness over the input space. As each prediction is a weighted average of known table points, the approach is also highly interpretable. On the other hand, the approach requires a sufficiently dense table to get good predictive accuracy, and the cost of table look-ups increases as the table density increases.

For model development and metaparameter tuning, we employed the kNNr implementation from the Python Scikit-Learn module[6]. For coupling to PFLOTRAN reservoir simulations, we relied on the open source FORTRAN code KDTREE 2 [15].

III.C. Artificial Neural Network

Artificial Neural Network (ANN) models are commonly employed by the machine learning community for regression and classification problems. They can be described as intricate networks of "artificial neurons" that are essentially weighted combinations of (usually simple) nonlinear functions. One motivation for the development of neural networks [6,7,10] was to create a regression approach for complex functions that avoids the combinatorial growth of the feature space that occurs in polynomial regression models.

ANN can be more accurate than kNNr using fewer training data as its functional representation helps to interpolate in areas where fewer training data are available. However, ANN models are not as readily interpretable as kNNr models, and care must be taken to avoid overfitting.

The ANN surrogate was developed in Python using the Tensorflow/Keras module[16]. A feed-forward neural network structure was selected with the popular rectified linear unit (ReLU) activation function. All training and metaparameter tuning was done in Python. For coupling to PFLOTRAN reservoir simulations, a Fortran ANN evaluator was written specifically for the selected network configuration. This evaluator reads in the ANN weights

that were determined offline in the Python training and tuning scripts.

IV. SURROGATES NOT RELYING ON DETAILED INTERNAL STATE INFORMATION

While the FMD model tracks detailed information about the internal state of the fuel cask over time, as listed in Table I, most of this information is not readily available when a surrogate model is called in a reservoir simulation. As such, our first approach to surrogate modeling considers only the following 6 features:

- Dose rate at fuel surface (= f (time, burnup))
- Temperature
- Environmental concentrations (CO_3^{2-}, O_2, Fe^{2+}, and H_2)

These features are either set by the environment (temperature and environmental concentrations) or can be readily computed from global information (dose rate).

As covered in detail in [17], surrogates using these features show good agreement with the process model predictions of the UO_2 fluxes.

TABLE III. Error metrics for kNNr and ANN surrogates on testing data for the case where no detailed internal fuel cask information is used.

Surrogate	nrmse	mape
kNNr	0.48	44%
ANN	0.52	25%

Table III shows the nrmse and mape errors on the UO_2 fluxes predicted by the kNNr and ANN surrogates compared to FMD simulations on testing data. In this case, the surrogate models were trained based on 400,000 FMD Matlab runs. The kNNr model used dose rate, temperature, and the concentrations of CO_3^{2-} and H_2 as features. The kNNr table contained 2.28 million samples and 80 Nearest Neighbors were used for interpolation. The ANN model used two hidden layers with 64 nodes each for a total of 4673 parameters, and was trained on the same data set using dose rate, temperature, and all 4 environmental concentrations CO_3^{2-}, O_2, Fe^{2+}, and H_2 as features.

A demonstration on a full-scale shale repository reference case simulation showed that the ANN and kNNr surrogate models enable accounting for more detailed FMD dynamics than when a Fractional Dissolution Rate approximation is used, while keeping the computational cost of reservoir simulations manageable [17].

V. SURROGATES USING CORROSION LAYER THICKNESS AS A FEATURE

V.A. Approach

While the results in [17] are encouraging, the accuracy of the surrogates is not superb. In this paper, we explore the potential of getting more accurate surrogate models by incorporating additional information about the internal fuel cask state. One variable that captures a lot of information about the amount of waste form degradation is the Corrosion Layer Thickness (CLT). Since this feature is not readily available without running a detailed FMD process model, a dual surrogate model approach is followed.

A first surrogate model predicts the CLT at the current time, using the CLT at the previous time step and the time step size as features, in addition to the features used in Section IV. A second surrogate predicts the UO_2 flux, using this same expanded feature set. After advancing to the next time step, the CLT predicted by the surrogate in the previous time step becomes part of the features for the next time step.

V.B. Preliminary Results

This dual surrogate approach was implemented for the kNNr surrogate, using dose rate, temperature, and the concentrations of CO_3^{2-} and H_2 along with CLT at the previous time step and the time step size as features. The surrogate was trained using a data set of 1 million FMD Matlab runs sampled from the distributions listed in Table II. After removing unphysical runs, 15% of the data was split off as validation data and 10% was split off as testing data, resulting in about 9.4 million validation data points, 6.3 million testing data points, and 47 million training data points. Following [17], the training data was downsampled by randomly selecting a number of samples from each FMD time trajectory.

Figures 2 – 5 show preliminary results that explore the choice of kNNr metaparameters (amount of training data and the number of Nearest Neighbors, NN) for the CLT and UO_2 Flux predictions. In all of these experiments, the training data was subsampled to 50 time samples per run. The validation data was used for computing the error metrics. As the CLT is initialized at a very small value on the order of 10^{-19} micrometer, the presence of some very small CLT values makes the computation of the regular mape error numerically unstable. To mitigate this effect, we defined a modified "floored" mape error metric, which uses a value of 10^{-4} in the mape denominator if the CLT at that point is less than 10^{-4} μm. This effectively removes the impact of division by very small CLT values from the computation. This modified mape metric here is indicated as mape_f in the figures.

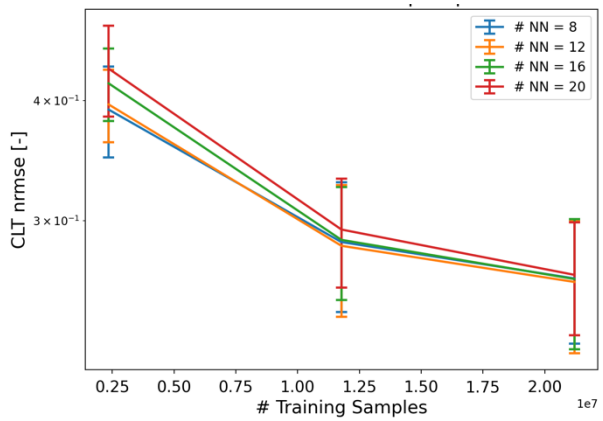

Fig. 2. The nrmse metric for prediction of the CLT as a function of the amount of training samples for different values of the number of Nearest Neighbors (NN) used.

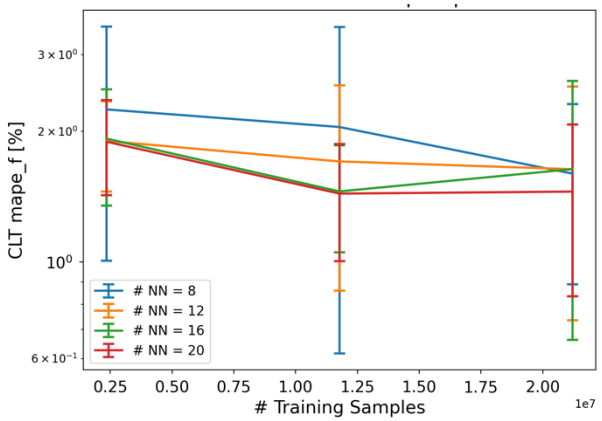

Fig. 3. The mape_f metric for prediction of the CLT as a function of the amount of training samples for different values of the number of Nearest Neighbors (NN) used. Errors are on the order of 1 – 2 %.

Figures 2 and 3 show that the incorporation of the CLT at the previous time step along with the time step size as features, allows a very accurate prediction of the CLT, with mape errors down to about 1 – 2% on the validation data.

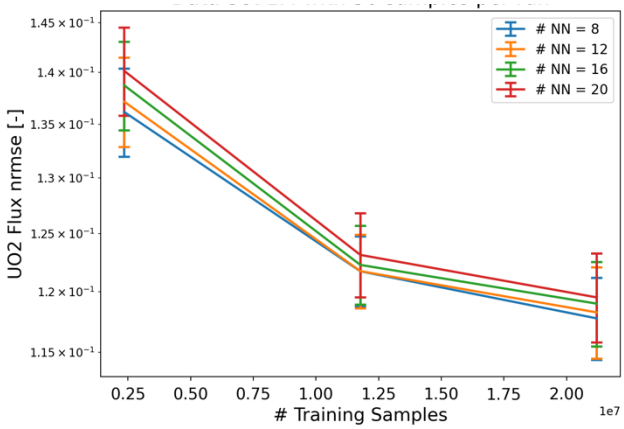

Fig. 4. The nrmse metric for prediction of the UO_2 flux as a function of the amount of training samples for different values of the number of Nearest Neighbors (NN) used.

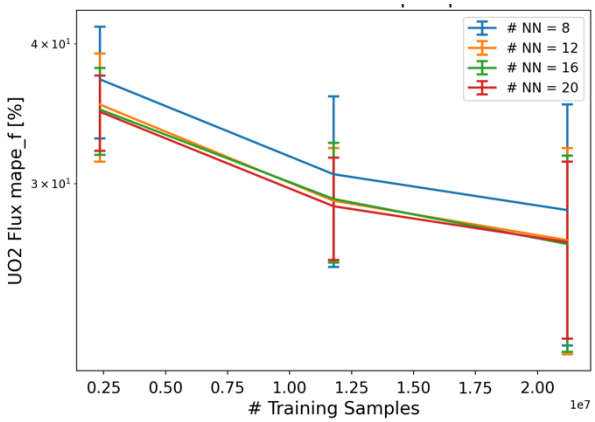

Fig. 5. The mape metric for prediction of the UO_2 flux as a function of the amount of training samples for different values of the number of Nearest Neighbors (NN) used. Given enough data points, the mape error is less than 30 %.

Figures 4 and 5 similarly show nice convergence of the kNNr surrogate approximation error in the UO_2 flux as more training data is provided.

Based on the trends in this preliminary tuning of the kNNr meta parameters, both the CLT and UO_2 flux are best predicted using about 8 – 12 nearest neighbors with as much training data as possible. Note that the number of nearest neighbors used here is much lower than the 80 nearest neighbors used in Section IV. As such, the addition of the CLT as a feature will allow for faster table lookups, and may also be more robust as there is less danger of grabbing points that are too far away when fewer neighbors are used in the interpolation. Based on these tuning results, a kNNr configuration of 10 nearest neighbors using all available training data (all 23 million samples from the data set that was downsampled to 50 samples per FMD process model run) was selected to predict the testing data. This testing data has not been used in any of the training and tuning of the kNNr surrogate.

Figures 6 and 7 below compare the kNNr predictions of the CLT and UO_2 flux to the testing data for 50 randomly sampled trajectories of the FMD process model. Note that in this comparison, each data point in the time trajectories was predicted on its own, using the features provided by the testing data. In a real case scenario, the true CLT value at the previous time step would not be available and would need to be approximated by the same surrogate. As such, errors in the surrogate approximations in previous time steps may result in compounding errors over time. The analysis shown here is still useful as it shows where such errors are most likely to originate.

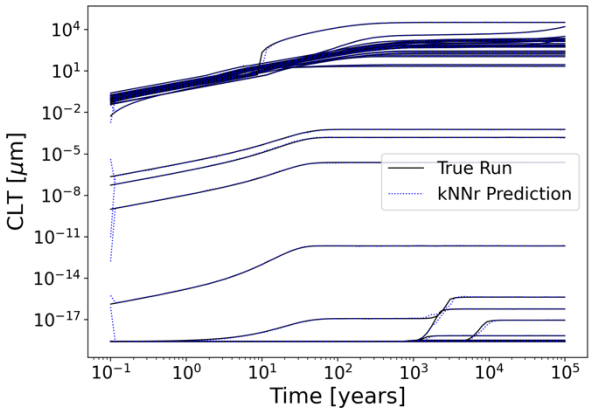

Fig. 6. Comparison of the True and kNNr prediction of the CLT for 50 randomly selected runs in the testing data.

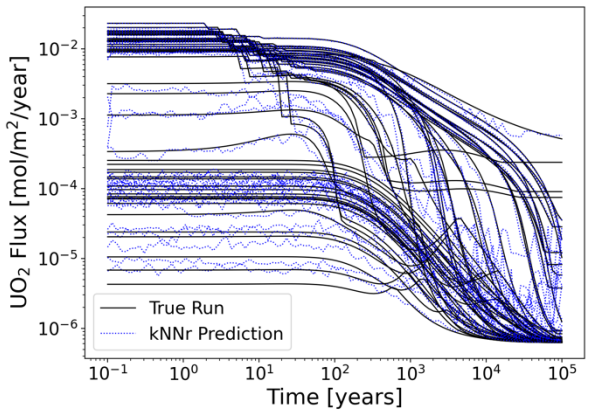

Fig. 7. Comparison of the True and kNNr prediction of the UO_2 flux for 50 randomly selected runs in the testing data.

Aside from some deviations early in time, the kNNr predictions of the CLT in Figure 6 are very close to the true values in the testing data. This graph also illustrates the very wide range in CLT values. It is believed that the errors in the CLT predictions at early time, when the CLT is often very small, are the ones that tend to inflate the mape error if no floor value is applied.

The predictions of the UO_2 fluxes in Figure 7 show good agreement with the test data, although the agreement is not as good as for the CLT predictions. As observed also in [17], the kNNr prediction is noisy as it is a local prediction, drawing information only from 10 nearest neighbors to each query point in the training sample space.

Overall, with this kNNr configuration of 10 nearest neighbors and 23 million training samples, the prediction of the UO_2 flux in the testing data shows an nrmse error of 0.11, and a mape error of 29%. Even without extensive tuning of the kNNr metaparameters, this is a significant improvement from the nrmse error of 0.48 and mape error of 44% reported in Table III for the case where no CLT information was used.

V.C. Ongoing Work

We are continuing to tune the kNNr meta parameters in order to optimize both the accuracy and speed of the predictions. In this process, we are also investigating the cause of the large (in relative sense) deviations in the CLT predictions that show up at early times. Besides kNNr, we will also train Artificial Neural Networks (ANNs) using the same feature set. As ANNs use a functional representation rather than local approximations, the predictions by the ANN surrogates are likely to be smoother than the kNNr predictions when looking at time trajectories. The resulting kNNr and ANN configurations will then be employed to full scale nuclear waste performance assessment simulations

VI. CONCLUSIONS

Two machine learning surrogate models are under development to rapidly emulate the effects of the Fuel Matrix Degradation (FMD) model in the *GDSA Framework*. One is a k-Nearest Neighbors regressor (kNNr) method that operates on a lookup table, and the other is an Artificial Neural Network. Both approaches have a high degree of accuracy, provided that enough training data is available with features that are informative of the UO_2 flux that results from the fuel degradation.

While earlier work [17], used only features that do not require detailed information about the internal state of the fuel cask, the current work explored the use of the Corrosion Layer Thickness (CLT). While this feature would need to be predicted along with the UO_2 flux at every time step in a reservoir simulation, the preliminary results in this work with the kNNr surrogate show that the CLT is very informative of the UO_2 flux. Including CLT as a feature therefore results in dramatically better accuracy. Ongoing work is incorporating the CLT as a feature in the ANN surrogate, and further refining the sampling schemes and meta-parameter tuning processes before employing this approach in realistic, full scale respository simulations.

The aim of these surrogate models is to enable the *GDSA Framework* to simulate spent fuel dissolution for each individual breached spent fuel waste package in a probabilistic repository simulation. Having the ability to emulate spent fuel dissolution in probabilistic PA simulations will have the added capability of allowing uncertainties in spent fuel dissolution to be propagated and sensitivities in FMD inputs to be quantified and ranked against other inputs.

ACKNOWLEDGMENTS

This work was funded by the US DOE, Office of Nuclear Energy, Spent Fuel and Waste Science and Technology.

Sandia National Laboratories is a multi-mission 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-NA-0003525.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. SAND2022-#### C.

REFERENCES

1. SNL. *GDSA Framework: A Geologic Disposal Safety Assessment Modeling Capability*. Available from: pa.sandia.gov (2017).
2. Jerden, J., J.M. Copple, K.E. Frey, and W. Ebert, *Mixed Potential Model for Used Fuel Dissolution - Fortran Code*, FCRD-UFD-2015-000159, US Department of Energy, Washington, DC (2015).
3. Mariner, P.E., W.P. Gardner, G.E. Hammond, S.D. Sevougian, and E.R. Stein, *Application of Generic Disposal System Models*, FCRD-UFD-2015-000126, SAND2015- 10037 R, Sandia National Laboratories, Albuquerque, New Mexico (2015).
4. Jerden, J., G. Hammond, J.M. Copple, T. Cruse, and W. Ebert, *Fuel Matrix Degradation Model: Integration with Performance Assessment and Canister Corrosion Model Development*, FCRD-UFD-2015- 000550, US Department of Energy, Washington, DC (2015).
5. Simpson, T.W., V. Toropov, V. Balabanov, and V. F.A.C. *Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not*. in *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2008. Victoria, British Columbia, Canada. AIAA Paper 2008-5802.
6. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, 2825-2830 (2011).
7. Ben-David, S. and S. Shalev-Shwartz, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, United Kingdom: Cambridge University Press (2014).
8. Xiu, D., *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press (2010).
9. Ghanem, R. and P. Spanos, *Stochastic Finite Elements: A Spectral Approach*. New York, New York: Springer Verlag (2002).
10. Rasmussen, C.E. and C.K.I. Williams, *Gaussian Processes for Machine Learning*. MIT Press (2006).
11. Santner, T., B. Williams, and W. Notz, *The Design and Analysis of Computer Experiments. New York, New York: Springer (2003).*
12. Storlie, C.B., L.P. Swiler, J.C. Helton, and C.J. Sallaberry, "Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models," *Reliability Engineering & System Safety*, 94(11), 1735-1763 (2009).
13. Viana, F.A.C., R.T. Haftka, and V. Steffen, "Multiple surrogates: how cross-validation errors can help us to obtain the best predictor," *Structural and Multidisciplinary Optimization*, 39(4), 439-457 (2009).
14. Helton, J.C. and F.J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Reliability Engineering & System Safety*, 81(1), 23-69 (2003).
15. Kennel, M.B., *KDTREE 2: Fortran 95 and C++ software to efficiently search for near neighbors in a multi-dimensional Euclidean space*, https://arxiv.org/pdf/physics/0408067.pdf, Institute for Nonlinear Science, University of California (2004).
16. Abadi, M., et al., *"TensorFlow: A System for Large-Scale Machine Learning",* 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16), Savannah, GA, November 2–4, 2016.
17. Debusschere, B.,J., D.T. Seidl, T.M. Berg, K.W. Chang, R.C. Leone, L.P. Swiler, P.E. Mariner, *Machine Learning Surrogates of a Fuel Matrix Degradation Process Model for Performance Assessment of a Nuclear Waste Repository*, Nuclear Technology, in preparation, (2022).