Adaptive Sampling of Dynamic Scenarios close to the Limit Surface using Deep Neural Network and Monte Carlo Dropout

Junyong Bae, Jong Woo Park, and Seung Jun Lee*
Ulsan National Institute of Science and Technology, 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Republic of Korea
*junyong8090@unist.ac.kr, jonwoo822@unist.ac.kr, sjlee420@unist.ac.kr**

1. Introduction

Probabilistic safety assessment (PSA) is widely used to evaluate and investigate the risk of a nuclear power plant (NPPs). This methodology combines event tree and fault tree models to efficiently identify the event sequences that fail in the ultimate safety goal and assumes a *static* probability of failure. However, this *static* PRA can be limited. For instance, there is only a Boolean representation of system success/failure while the partial operation of components can induce a different consequence. Also, the timing of events is rigidly preset by the analyst, therefore, not considerably modeled.

Dynamic PSA has been suggested to supplement these flaws. In general, dynamic PSA takes count into t partial operations of components and systems, and time-dependencies between events to realistically model target system response. Therefore, Dynamic PSA generates numerous sequences which are originally represented by the few sequences in classical PSA. It implies that expensive deterministic analysis including thermal-hydraulic (TH) code runs should be iteratively executed to analyze the scenarios of Dynamic PRA. To overcome this computational challenge, research has been conducted to replace the expensive code runs with a simplified surrogate model.

A data-driven model has been widely studied as a surrogate model of the expensive TH code. Radaideh et al. predicted the sequences of important parameters, such as cold-leg temperature and peak cladding temperature (PCT), using deep neural networks under small break loss of coolant accident (LOCA) [1]. Deep learning-based accident trend estimation (DeBATE) has been developed by the Korea Atomic Energy Research Institute, which can estimate the trends of important parameters using a quantile recurrent neural network [2]. The feasibility of the simplified DeBATE models was validated by training 8,000 input-output data sets calculated by MARS-KS (Multi-dimensional Analysis of Reactor Safety KINS Standard, i.e., the Korean regulatory safety analysis code) [3, 4]. However, a data-driven model can generate strange predictions when untrained scenarios are given. Since a data-driven approach does not model thermal-hydraulic phenomena, the strange predictions are unreliable.

Nonetheless, a data-driven approach can focus computational resources on the region of interest. A limit surface/state (LS) is a practical concept for system reliability assessment. In the scenario consequence space spanned by scenario configuring parameters, the LS discriminates the system failure and success regions. Therefore, if the LS can be identified, the result of not-simulated scenarios can be conservatively assumed. Therefore, an algorithm finding the LS with a minimized number of simulations can save computation resources while retaining risk-sensitiveness and informativeness. A data-driven model can be a metamodel for this algorithm.

The Idaho National Laboratory developed a limit surface searching algorithm with adaptive sampling and an active learning method. This algorithm samples not-simulated scenarios based on the predictions of the metamodel, which trains the record of previous simulations [5]. They tested various kinds of metamodels such as support vector machines (SVMs) [6], k-nearest neighborhood [7], and Gaussian process [5]. Similarly, adaptive kriging Monte Carlo simulation (AK-MCS) has been applied to identify operational conditions that lead to failure of a passive safety system and a lead fast reactor, respectively [8, 9]. AK-MCS embeds kriging, i.e., Gaussian process modeling, as a metamodel to estimate the consequence of not-simulated scenarios [10].

This research suggests an adaptive sampling method to identify LS with a minimized number of simulations using deep learning and Monte Carlo dropout (MCDO). The suggested method is analogous to AK-MCS; however, we embed a different metamodel (i.e., deep neural network) and tailored it to be specialized in finding the LS. Retained advantaged is that the suggested method and AK-MCS consider not only closeness to the LS but also the uncertainty of consequence predictions. To investigate the prediction uncertainty, we employed the MCDO method introduced by Gal and Ghahramani in 2016 [11]. The case study result shows that the suggested sampling method can efficiently find out the LS with a minimized number of simulations.

2. Background concepts

2.1. Deep neural network

A deep neural network, also called multilayer perceptrons, is a basic form of deep learning model. It consists of massive connections between logical units. In general, logical units composes a layer, and layers are stacked sequentially. Since this structure has depth, this structure is called a *deep* neural network. Thanks to this depth, it can approximate complicated functional relationships between inputs and outputs. The training of a neural network is to coincide the network outputs with the desired outputs by adjusting network parameters such as connection weights between logical units. This adjustment is conducted by a backpropagation algorithm which spreads out the deviation between the network outputs and the desired outputs from an output layer in a backward direction. Currently, based on dramatically improved computation power and efficient open-source software library (e.g., Tensorflow and PyTorch), deep neural networks have been solving problems in various areas.

2.2. Monte Carlo Dropout (MCDO)

In many deep learning tasks, quantifying the assurance or uncertainty of the predictions can be useful. For instance, in the case of accident diagnosis of an NPP, if a DNN model diagnosis the current accident with high uncertainty, human operators can take over the diagnosis task from the network model. However, most deep learning models only provide the deterministic label and values for classification and regression, respectively, without any information about uncertainty or assurance.

As an uncertainty quantification method for DNN, an MCDO is to randomly detach logical units from the network as shown in fig. 1. This detachment called dropout is originally introduced as a regularization technique to prevent overfitting of the deep learning model.

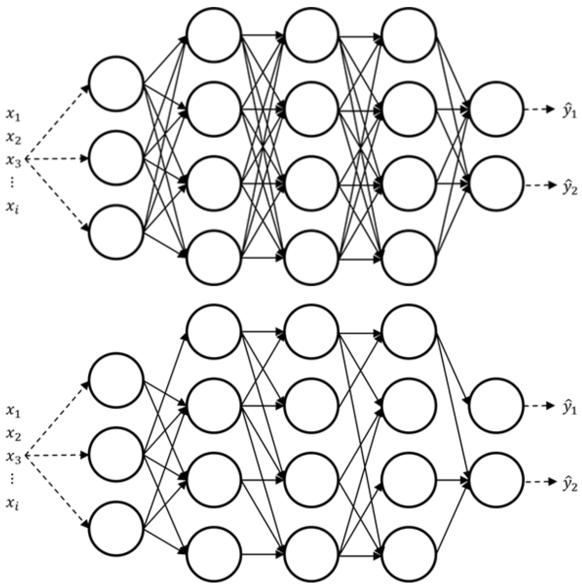

Fig. 1. The structure of a simple neural network without and with dropout.

However, Gal and Ghahramani found that prediction uncertainty can be obtained by simply activating the dropout not only in training but also in the testing phase [11]. Implementation of MCDO is simple: repeatedly predict the outputs with the same input and different dropout configuration and use the mean and variance of the outputs as the prediction and variance, respectively. Intuitively, when the input is not like the ones in training data sets, the MCDO outputs will be high variational. MCDO does not produce high-quality prediction uncertainty, however, is easy to implements and requires training of a single neural network. Figure 2 shows the example of MCDO for two data points with low uncertainty (upper figure) and high uncertainty (below figure).

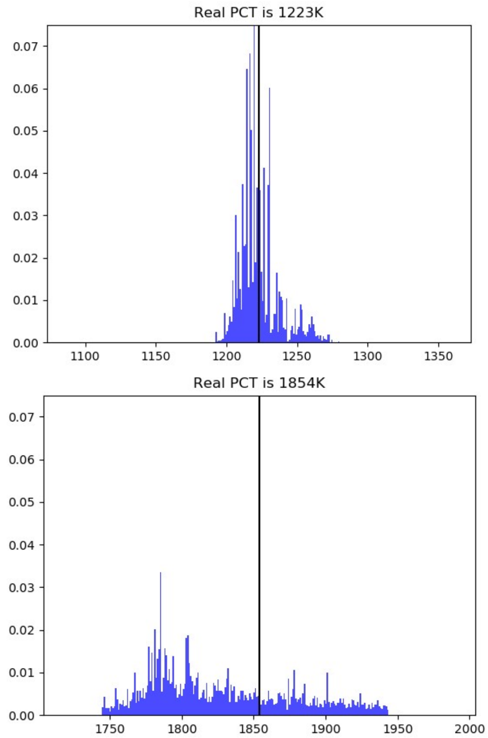

Fig. 2. MCDO results of the deep neural network predicting PCT.

3. Adaptive sampling for limit surface search

We suggest an adaptive sampling method for LS search that uses as few simulations as possible based on the DNN metamodel that predicting the consequence-predicting model. For instance, the DNN can be designed to predict the PCT when the performance and timing of safety injection are given. As an adaptive sampling and active learning process, this method is an iterative process that samples the scenarios that are estimated to be close to LS by the DNN, simulates the sampled scenarios, trains the DNN with the updated simulation record, and samples the scenarios again. It is important to note that our sampling method selects the scenarios to be simulated based on not only adjacency to the LS, but also relative uncertainty, as shown in Eq. 1. The followings are detailed steps of our method:

1. Defines the space spanned by the scenarios configuring parameters and initializes the DNN.

2. Simulates extreme scenarios (when scenarios configuring parameters are the maximum or minimum value in uncertain domains) and trains the DNN.
3. Conducts MCDO for all scenarios and derives means and variances.
4. Evaluate the score S_k of each scenario with Eq. 1.
5. Samples N scenarios randomly among the scearnarios satisfying $S_k < D$. N and D are hyperparameters that should be predefined.
6. Simulate sampled configurations and train the DNN regressor
7. Check convergence. If not, go back to step 3

$$S_k = \left| \frac{Mean_k - Failure\ Criterion}{Variance_k} \right| \quad (1)$$

4. Case study

To verify the effectiveness and feasibility, the suggested sampling method was applied to find the LS in the dynamic scenario space when a large break loss of coolant accident (LOCA) happens. For simplicity, we supposed only two safety systems: safety injections tanks (SITs) and low power safety injection (LPSI). Dynamic scenarios were generated by assuming partial operations of three SITs and LPSI and delayed generation of safety injection actuation signal (SIAS) due to malfunction of ESFAS. There were five performances of three SITs (0%, 25%, 50%, 75%, and 100%), 23 performances of LPSI (100%, 92%, 83%, 79%, 75%, 71%, 67%, 63%, 58%, 54%, 50%, 46%, 42%, 38%, 33%, 29%, 25%, 21%, 17%, 13%, 8%, 4%, and 0%), 14 delayed times of SIAS generation (0s, 30s, 60s, 90s, 120s, 150s, 180s, 210s, 240s, 270s, 300s, 330s, 360s and 350s), generating 40,250 dynamic scenarios with different configuration of five factors. As validation data sets, the scenarios were simulated by thermal-hydraulic system code and labeled as core damage (CD) when the PCTs exceed 1478 K.

The hyperparameter N and D were set by 403 and 1, respectively. The metamodel DNN consists of three hidden layers composed of 64 artificial neurons and is compiled with loss function of mean squared error (MSE) and Adam optimizer with a learning rate of 0.001. Figures 3, 4, and 5 show the change of metamodel predictions for each iteration. To display the LS in five-dimension, we assumed the performances of SITs. The red-colored area is the region where the metamodel predicts CD and vice versa. The dots represent the scenarios selected by the sampling method and denoted numbers are real values of the PCTs.

Regarding Fig. 3, the metamodel initially gives the inaccurate prediction when trains extreme case only s (Iteration 0). Nonetheless, it improves the predictions by training the simulation results of the adaptively sampled scenarios (Iteration 1~26). As shown in the figure of the last iteration (Iteration 26), most of the sampling scenarios are close to the LS and have PCTs close to 1478 K. This tendency can be verified in both Fig. 4 and 5. In the case of Fig. 4, only a few scenarios are sampled initially (Iteration 3) and no more sampling since the estimated PCTs are too high. In contrast, in the case of Fig. 5, most of the scenarios in non-CD regions are sampled because the estimated PCTs are close to 1478 K. Results shown in Figs. 3, 4, and 5 indicate that the suggested method can adaptively sample the scenarios that are close to the LS.

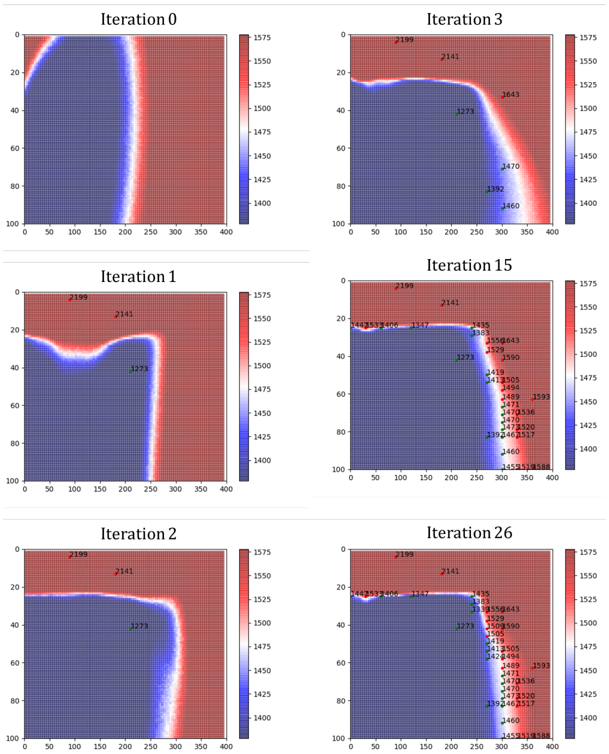

Fig. 3. The change of metamodel predictions when SITs performances are (SIT1, SIT2, SIT3 = 50%, 50%, 0)

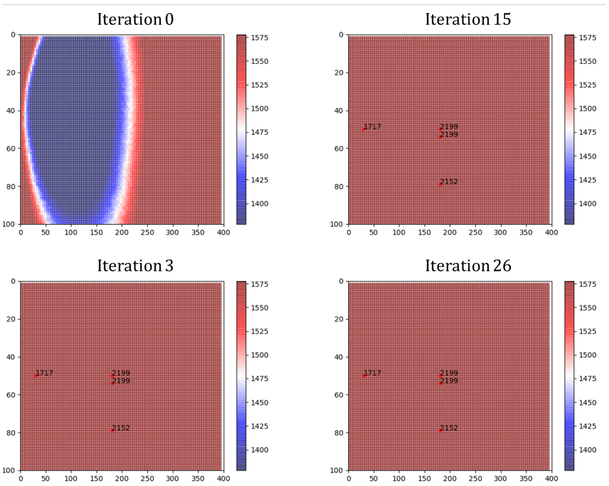

Fig. 4. The change of metamodel predictions when SITs performances are (SIT1, SIT2, SIT3 = 0%, 25%, 0)

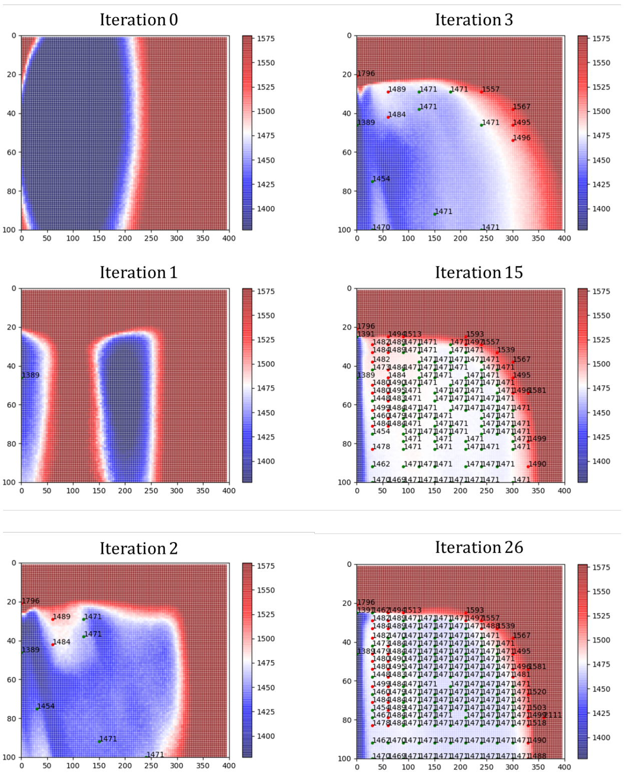

Fig. 5. The change of metamodel predictions when SITs performances are (SIT1, SIT2, SIT3 = 25%, 25%, 0)

Figure 6 shows the number of the sampled scenarios for each iteration and the classification accuracy as the number of sampled scenarios. Since the last iteration samples all suspected scenarios, a sharp increase of sampled scenarios is observed. The sampling method consequently sampled 4,758 scenarios (11.8% of all dynamic scenarios) while achieved classification accuracy of 99.8% (17,691/17,756 for non-CD and 22,479/22,494 for CD scenarios). Note that sampled scenarios were classified according to the results of the simulation and not-sampled scenarios were classified according to the prediction of the metamodel.

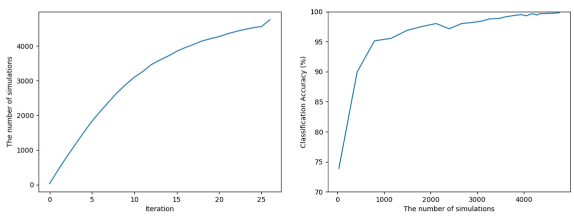

Fig. 6. The accumulated number of simulations for each iteration (left), and classification accuracy for each iteration (right)

5. Conclusion

In this research, we suggest an adaptive sampling method for the LS search while minimizing the simulations and conserving informativeness and risk-sensitiveness. The case study shows that how the sampling method explores the dynamic scenario spaces spanned by the scenario configuring parameters. The simulations were emphasized on the LS. The trained metamodel also could make conservative assumptions about not-simulated scenarios. Therefore, we believed that the suggested sampling method can be utilized with system reliability analysis and Dynamic PSA as guiding tools for expensive thermal-hydraulic system code runs. Furthermore, this algorithm can designate an instance that sophisticated analysis with high resolution is necessary. For both applications, our research could significantly reduce computational costs and time.

Further study will be conducted to check the sensitivity of the suggested method. For instance, as hyperparameters, N and D can significantly change the behavior of the sampling process. For instance, if lower N (i.e., less simulation for each iteration) and higher D (i.e., more scenarios satisfy $S_k < D$) are given, the sampling will be more delicate, however, the number of simulation and computational efforts for DNN training and MCDO will also increase. In addition, integration with PSA should be researched.

REFERENCES

[1] M.I. Radaideh, C. Pigg, T. Kozlowski, Y. Deng, A. Qu, Neural-based time series forecasting of loss of coolant accidents in nuclear power plants, Expert Systems with Applications, 160 (2020) 113699.
[2] G. Heo, S. Baek, D. Kwon, H. Kim, J. Park, Recent research towards integrated deterministic-probabilistic safety assessment in Korea, Nuclear Engineering and Technology, (2021).
[3] H. Kim, J. Cho, J. Park, Application of a Deep Learning Technique to the Development of a Fast Accident Scenario Identifier, IEEE Access, 8 (2020) 177363-177373.
[4] B.D. Chung, K.D. Kim, S.W. Bae, J.J. Jeong, S.W. Lee, M.K. Hwang, MARS code manual volume II: input requirements, in, Korea, Republic of, 2010, pp. 344.
[5] A. Alfonsi, C. Rabiti, D. Mandelli, J.J. Cogliati, C. Wang, D.P. Maljovec, P.W. Talbot, C.L. Smith, Raven theory manual, in, Idaho National Lab.(INL), Idaho Falls, ID (United States), 2016.
[6] D. Mandelli, C. Smith, Adaptive sampling using support vector machines, in, Idaho National Laboratory (INL), 2012.
[7] D. Mandelli, A. Alfonsi, C. Smith, C. Rabiti, J. Cogliati, IMPROVED SAMPLING ALGORITHMS IN RISK-INFORMED SAFETY APPLICATIONS, (2015).
[8] P. Turati, A. Cammi, S. Lorenzi, N. Pedroni, E. Zio, Adaptive simulation for failure identification in the advanced lead fast reactor European demonstrator, Progress in Nuclear Energy, 103 (2018) 176-190.
[9] L. Puppo, N. Pedroni, A. Bersano, F. Di Maio, C. Bertani, E. Zio, Failure identification in a nuclear passive safety system by Monte Carlo simulation with adaptive Kriging, Nucl Eng Des, 380 (2021).
[10] B. Echard, N. Gayton, M. Lemaire, AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation, Struct Saf, 33 (2011) 145-154.
[11] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, PMLR, 2016, pp. 1050-1059.