Deep-learning for Guided Simulation of Scenarios for Dynamic Probabilistic Risk Assessment

Junyong Bae [a], Jong Woo Park [b], and Seung Jun Lee [c]
[a] Department of Nuclear Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, Republic of Korea, [a]junyong8090@unist.ac.kr, [b] jongwoo822@unist.ac.kr, [c] sjlee420@unist.ac.kr

Abstract: One of the practical challenges of simulation-based dynamic risk assessment is to minimize the number of simulations that impels computationally expensive code runs such as thermal-hydraulic system codes. To tackle this challenge, this research introduces a guided simulation algorithm inspired by a human reasoning process. This approach utilizes deep learning methods including a deep neural network and Monte-Carlo dropout. A deep neural network is employed as a surrogate model predicting the consequences of the postulated scenarios based on the code run results, and a Monte-Carlo dropout is applied to estimate the predicting confidence. These predicted consequences and their confidence guide a user to sample and simulate the scenarios close to the boundary intensively, and ultimately, to identify the success or failure of all scenarios with minimized code runs. We verified the applicability of the suggested approaches with data sets of simulation results of a loss of coolant accident scenarios. These deep learning approaches could be utilized as a simulation optimizing engine for an advanced dynamic risk assessment framework, alongside a probability-based optimizing framework.

1. INTRODUCTION

Over the past decades, various research has suggested dynamic methodologies to supplement static assumptions of a probabilistic risk assessment (PRA). These methodologies attempt to enhance the realism of PRA results by considering the possible temporal and partial behavior of plant operators and components. To this end, they divide a single scenario in static PRA into several scenarios based on the dynamic behavior of the operators and components, which means the number of scenarios that should be deterministically analyzed by a physical model, such as a thermal-hydraulic (TH) system codes, dramatically increased. To effectively address these scenarios, the currently developed ones, such as analysis of dynamic accident progression trees (ADAPT) [1], risk analysis virtual environment (RAVEN) [2, 3], and dynamic integrated consequence evaluation (DICE) [4, 5], provide an environment where a probabilistic scenario generator is integrated with a deterministic consequence analyzer using the physical model.

One of the practical challenges of these simulation-based dynamic PRA is that the consequences of an enormous number of scenarios should be identified by the physical model runs, which are computationally expensive. This challenge is also existing in other domains such as a structural reliability assessment. In this domain, the most common approach for estimating system reliability is the Monte Carlo Simulation (MCS). MCS estimates the probability of failure as the ratio of the failure scenarios among the random population sampled by the predefined distributions of scenario configuring parameters. If a system failure is a rare event, this crude approach inevitably requires a large random population of scenarios to ascertain the estimation, consequently, impels the excessive executions of a complex engineering model [6]. To tackle this problem, Echard et. al. proposed an active learning reliability method combining Kriging and Monte Carlo Simulation (AK-MCS) [6]. This method aims to minimize the execution of the engineering model by predicting a consequence (i.e., success or failure) of each scenario with a metamodel. AK-MCS is active learning and adaptive sampling process for the metamodel and scenarios, respectively, where the scenarios close to a decision boundary (i.e., limit surface) are sampled and simulated, the training data sets are updated by the simulation results, and the metamodel is trained and locates the boundary, iteratively. The metamodel trained through this process becomes robust against the narrowest success/failure scenario, therefore, it can accurately predict the outcome of each scenario and ultimately estimate the probability of failure. The metamodel of AK-MCS is a kriging model (i.e., Gaussian process regression) which generates predictive uncertainty along with the prediction. Taking this advantage, AK-MCS improves the sampling efficiency by prioritizing the scenarios with not only closeness to the boundary but also high predictive uncertainty. AK-MCS has been applied to a safety assessment of a nuclear power plant. Puppo et al. employed AK-MCS to identify the operational conditions of a passive safety system that cause unsafe conditions [7, 8]. Turati et al. also employed this method to explore accident scenarios of a lead fast reactor (LFR) [9].

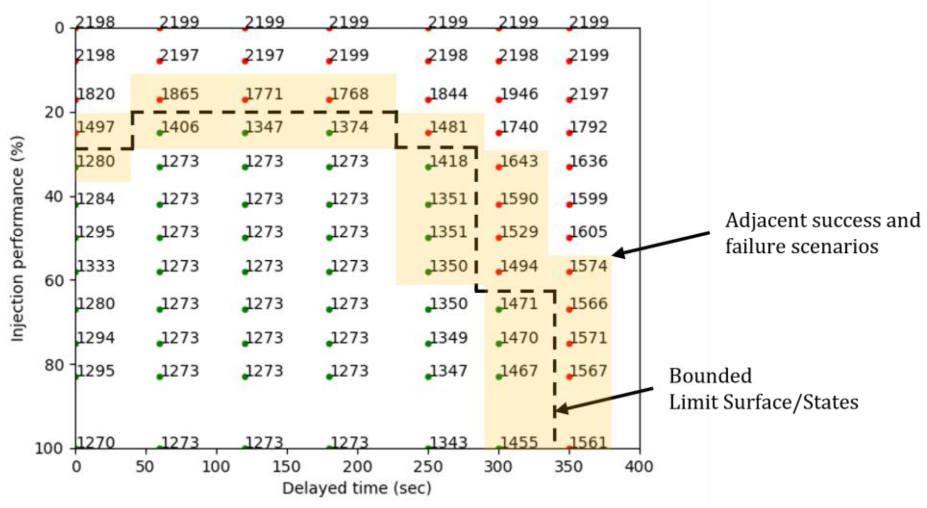
AK-MCS can be applied to a dynamic PRA since it can minimize the number of simulations by locating a decision boundary. For this, a different metamodel should be employed. In contrast to MCS, the scenario population of a dynamic PRA consists of the scenarios with their probabilities. In this case, a decision boundary should be estimated by the simulation records of more than thousands of adjacent success and failure scenarios even if the failure is a rare event. However, the Gaussian process is hard to address more than thousands of data sets since it has a cubic time complexity $O(n^3)$ where n is the size of the training data sets [10].

This research introduces a guided simulation algorithm named Deep learning-based Searching Algorithm of Informative Limit Surface/State/Scenarios (Deep-SAILS) for a dynamic PRA [11-13]. Deep-SAILS is analog to AK-MCS, however, employs a deep neural network as the metamodel. In this algorithm, the deep neural network is a high-fidelity surrogate model that learns more than tens of thousands of data sets of simulation records and estimates the consequence of more than millions of scenarios. We also increase the scenario sampling efficiency by utilizing the predictive uncertainty of the deep neural network quantified by a Monte Carlo Dropout [14]. The feasibility study of Deep-SAILS was conducted by simulation records of dynamic scenarios of small break loss of coolant accident (LOCA) and, as a result, the algorithm identified the success or failure of each scenario with more than 99.9% accuracy while referring to the records of only 11.13% of total scenarios in average.

2. Deep-SAILS: GUIDED SIMULATION ALGORITHM WITH DEEP-LEARNING

For decades, selective simulations of interesting ones among a scenario population have been discussed to alleviate the computational cost necessitated by a dynamic PRA. Interesting scenarios depend on the purpose of the analysis; however, the widely accepted ones are the scenarios where the consequence is success or failure by a narrow margin since these scenarios can locate the limit surface/states (LS) between the success and failure regions and make a conservative assumption about the remaining scenarios. Figure 1 shows the example of simulation results of dynamic scenarios configured by the delayed time of an engineering safety function actuation signal (ESFAS) generation (x-axis) and percentile performance of safety injection (y-axis) under large break LOCA [13]. Success and failure scenarios are colored by green and red, respectively and the peak cladding temperature (PCT) of each scenario is denoted. In this case, the scenarios in the yellow shaded area can be of primary interest. If the results of these scenarios are given, we can reasonably assume the success or failure of the remaining ones, and therefore, save the cost for the physical model runs. This approach has been researched by the Idaho National Laboratory (INL). They adopted a metamodel including classifier models [3] such as support vector machine [15] and k-nearest neighborhood classifier [16] and developed an adaptive sampler that finds a limit surface and samples the scenarios close to the founded surface.

Figure 1: Simulation results of dynamic scenarios configured by the delayed time of ESFAS generation and percentile performance of safety injection [13]* .



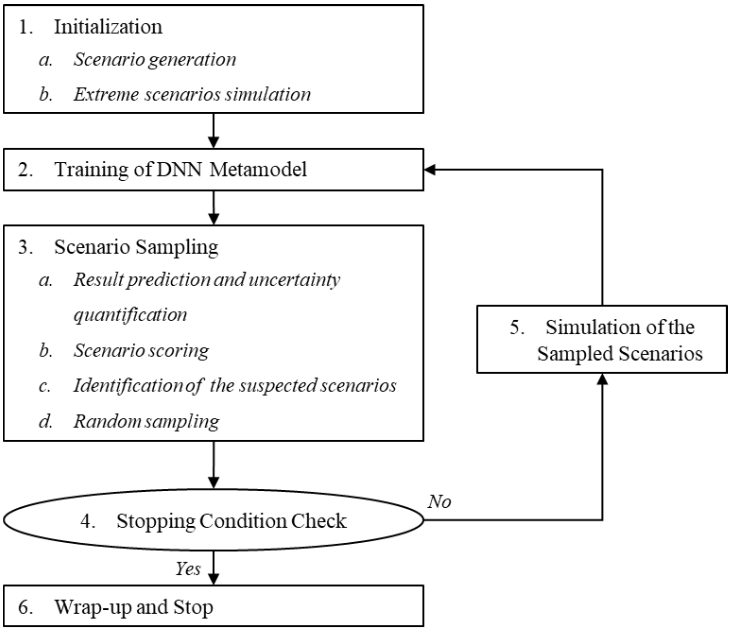
Deep-SAILS is in line with the previous research on simulation optimization for a dynamic PRA. The first purpose of Deep-SAILS is to simulate risk-sensitive scenarios (i.e., close to the limit surface) preferentially. To this end, the deep neural network predicts the consequences (e.g., PCTs) of all postulated scenarios and identifies the scenarios that are estimated to be close to the limit surface. The second purpose of the algorithm is to train the deep neural network to be robust for the scenarios close to the limit surface. This surrogate model may generate inaccurate predictions for the consequences of the scenarios far from the limit surface, however, the success or the failure of the scenario can be accurately identified because the model is aware of the limit surface. In other words, the trained deep neural network can establish conservative assumptions for non-simulated scenarios. Ultimately, a user of Deep-SAILS can figure out the success and failure of all postulated scenarios accurately with the minimized execution of a physical model.

Deep-SAILS is an iterative process of scenario sampling and neural network training. Figure 2 is a flow diagram of the algorithm [13]. It consists of five steps; initialization, training of the deep neural network, scenarios sampling, stopping condition check, and simulation of sampled scenario. The initialization includes a generation of a population of scenarios configured by the parameter (e.g., delayed time or performance of a safety system) and a preferential simulation of extreme scenarios as initial training data sets for the deep neural network. The extreme scenarios are the scenarios configured by the maximum and minimum values of each parameter. For instance, if three parameters are given, the number of the extreme scenarios is $2^3 = 8$.

After learning the consequences of the extreme scenarios, the network makes initial guesses about the consequences of non-simulated scenarios. Next, the algorithm samples the scenarios estimated to be close to the limit surface. The sampling method is the most important part of Deep-SAILS and is detailed in the next sections. With the sampled scenarios, the algorithm checks for a stopping condition. The stopping condition is satisfied when the proportion of already simulated scenarios out of the sampled ones exceeds a set point. Unless the stopping condition is satisfied, the algorithm simulates the sampled ones with the physical model and adds the simulation results to the training data sets of the network. The network is trained by the data sets and predicts the consequences of non-simulated scenarios again.

* Reprinted from Applied Soft Computing, 124, Junyong Bae, Jong Woo Park, Seung Jun Lee, Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment, Copyright (2022), with permission from Elsevier.

Figure 2: Flow diagram of Deep-SAILS [13]†



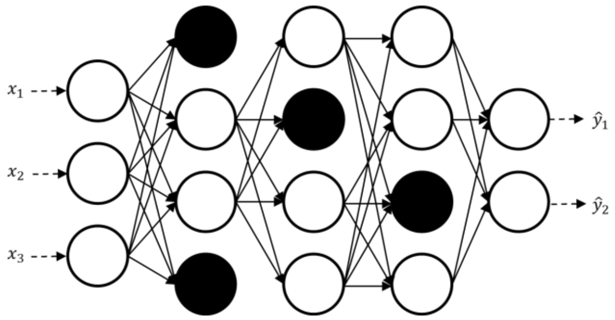
2.1. Deep neural network and Quantification of Predictive Uncertainty

A neural network is a set of interconnected logical units. When a plurality of units is interconnected and stacked, the network has a deep structure and can solve a complex problem. With advanced logical units (e.g., recurrent neural network) and connection designs (e.g. attention algorithm), a deep neural network is widely being applied in a nuclear field [17-19]. The logical units of a neural network are connected to each other according to their weights. Based on the training data sets, the weights are adjusted to achieve the desired output for a given input.

A deep neural network can generate a poor performance when it overfits trained cases and its outputs are varied only by some weights. Regularization techniques such as batch normalization and a dropout aim to mitigate this overfitting problem by training the weights uniformly. Especially, dropout is most widely used due to its simplicity and effectiveness [20]. For each training trial, it randomly omits some weights and adjusts the remaining weights, as shown in the below figure [13]. Once trained, a deep neural network infers with all weights.

† Reprinted from Applied Soft Computing, 124, Junyong Bae, Jong Woo Park, Seung Jun Lee, Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment, Copyright (2022), with permission from Elsevier.

Figure 3: Example of a deep neural network with a dropout [13][‡]. The weights for the black-colored nodes are omitted.



The research conducted by Gal and Ghahramani showed that a deep neural network with a dropout can approximate the Gaussian process [14, 21]. If a dropout is activated in the inference phase, a network produces different outputs depending on the dropout configurations. The means and variance of these outputs can be interpreted as the prediction and uncertainty, respectively. This uncertainty is strongly influenced by the number of data sets similar to the given input. This process with a dropout is named Monte Carlo Dropout.

Deep-SAILS utilizes a deep neural network as a surrogate model and quantifies the predictive uncertainty using MCDO. For a given scenario, the deep neural network predicts the consequences (e.g., PCT) multiple times with a random dropout configuration. Since this process is iterative for every scenario, this process should be implemented by efficient machine learning libraries such as Tensorflow or PyTorch and accelerated by the devices optimized for vectorized computations.

2.2. Scenario Sampling with *U-learning function*

Deep-SAILS scores each scenario by the *U-learning function*, suggested by the AK-MCS [6]. The *U-learning function*, $U(X_i)$, is defined in Equation (1), where X_i is scenarios configuration, $\hat{y}_i$ and $\sigma_{\hat{y}_i}$ are consequence prediction and uncertainty, respectively, given by the deep neural network, and a is a failure criterion (e.g., PCT limitation)

$$U(X_i) = \frac{|\hat{y}_i - a|}{\sigma_{\hat{y}_i}} \tag{1}$$

This function gives a lower score for the scenarios where the estimated results are closer to the failure condition (i.e., the numerator) and have higher uncertainty (i.e., the denominator). Especially, the denominator helps the algorithm sample the scenarios more meticulously since the given scenarios will be prioritized when there are no simulation results of the scenarios adjacent to the given one. The sensitivity study with or without the denominator can be found in the previous research [6, 12, 13].

Next, the scenarios with scores lower than the predefined distance, D, are identified as suspicious scenarios. The distance, D, is a critical hyperparameter determining the behavior of the algorithm. Lastly, N scenarios are randomly sampled from the suspicious population to encourage exploration

[‡] Reprinted from Applied Soft Computing, 124, Junyong Bae, Jong Woo Park, Seung Jun Lee, Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment, Copyright (2022), with permission from Elsevier.

3. CASE STUDY

3.1. Dynamic Scenario SLOCA

As a case study, Deep-SILAS was applied to the dynamic scenarios of small break LOCA [13, 22]. We used a TH system code that replicates the behavior of the Zion nuclear power plant, a representative Westinghouse 4-loop pressurized light water reactor with an electric power generation of 1000 MW. We assumed a 2-inch break (1.86×10^{-3} m^2) in the one cold leg and the dynamic behavior of two safety systems: High-pressure safety injection (HPSI) and atmospheric dump valve (ADV). For HPSI, delayed time of HPSI actuation and degraded performance are considered. For ADV, delayed open time is considered. The uncertain domain and discretization of these time and performance are detailed in Table 1. Consequently, 10,143 dynamic scenarios are generated. For efficiency, we simulated these scenarios ahead. A scenario is classified as the core damaged (i.e., failure) when PCT exceeds 1478K and as a success when vice versa.

Table 1: Uncertain domains and discretization of scenario configuring parameters under small break LOCA [13][§].

Parameter	Unit	Uncertain domain	Discretization
HPSI delayed time	min	(0, 60)	21 (0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60)
HPSI performance	min	(0, 60)	21 (0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60)
ADV open time	%	(0, 100)	23 (100, 92, 88, 83, 79, 75, 71, 67, 63, 58, 54, 50, 46, 42, 38, 33, 29, 25, 21, 17, 13, 8, 0)

3.2. Results

We set the distance D as 2.0 and N as 101 (i.e., 1% of the number of dynamic scenarios). The proportion for the stopping condition was assumed as 0.9. We executed the Deep-SAILS 10 times and Table 2 shows the results. The algorithm scored more than 99.98 % accuracy while simulating only 11.13 % of all postulated scenarios.

[§] Reprinted from Applied Soft Computing, 124, Junyong Bae, Jong Woo Park, Seung Jun Lee, Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment, Copyright (2022), with permission from Elsevier

Table 2: Classification results for each execution of Deep-SAILS.

Execution	Simulated Scenarios	Treu result = Success		Treu result = Failure		Error rate
		Classified as Success	Classified as Failure	Classified as Failure	Classified as Success	
1	1040	3772	5	6362	4	0.09 %
2	1187	3777	0	6366	0	0 %
3	1116	3774	3	6365	1	0.04 %
4	1232	3777	0	6365	1	0.01 %
5	1140	3776	1	6364	2	0.03 %
6	1129	3776	1	6365	1	0.02 %
7	1159	3776	1	6366	0	0.01 %
8	1119	3776	1	6365	1	0.02 %
9	1067	3777	0	6365	1	0.01 %
10	1099	3777	0	6366	0	0 %
Average	1129 (11.13%)	3776	1	6365	1	0.02 %

Figures 4 and 5 show the scenario sampling process and estimation of the deep neural network for iterations of execution 1 in Table 2. The x and y-axis are a delayed time of HPSI and ADV open time, respectively, and HPSI performance was assumed to be 75%. The green and red dots represent the sampled success and failure scenarios, respectively. The background colors also represent the success (blue-colored region) and failure region (red-colored region) and the limit surface (white-colored region) estimated by the trained deep neural network for each iteration. As shown in iteration 19 (last iteration) in figure 5, the majority of the sampled scenarios lie on the limit surface.

Figure 4: Sampled scenarios (green and red dots) and the LS (white region) pinpointed by the DNN metamodel for iterations 0, 1, 2, and 6 when HPSI performance is 75%

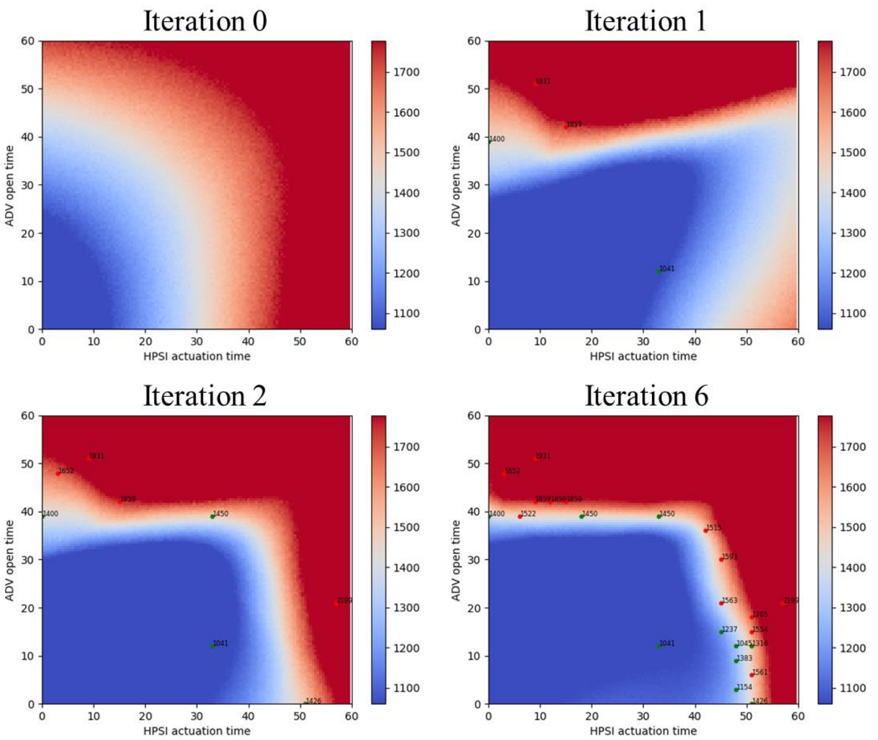
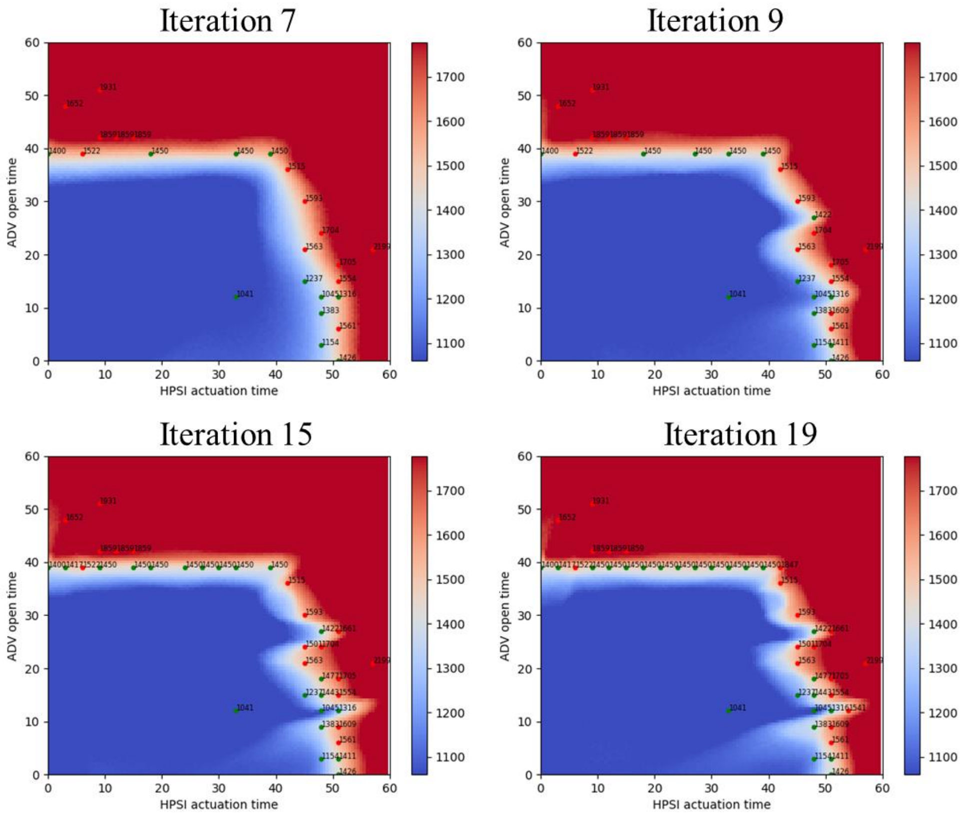


Figure 5: Sampled scenarios (green and red dots) and the LS (white region) pinpointed by the DNN metamodel for iterations 7, 9, 15, and 19 when HPSI performance is 75%



Both figures also show how the algorithm found the limit surface. With simulation results of the extreme scenarios, the network made a loose guess about the surface (Iteration 0 in figure 4). After several iterations, the network successfully found a rough location of the limit surface (Iteration 6 in figure 4). From this point, the algorithm exploited the limit surface and corrected the details of the surface. Comparing the plots of iteration 7 and iteration 19, the boundary when HPSI delayed time is roughly 40 ~ 50 min became more sophisticated.

4. CONCLUSION

This research introduced a guided simulation algorithm of a dynamic PRA, named Deep-SAILS. To overcome the limitation of previous including AK-MCS and an adaptive sampler, the algorithm employed a deep neural network as a high-fidelity surrogate model. In addition, the algorithm retains the strong point of AK-MCS, that is utilization of uncertainty information for a meticulous sampling of scenarios. To this end, the predictive uncertainty is quantified by the Monte Carlo Dropout technique.

This algorithm aims to sample and simulate the scenarios close to the limit surface and train a deep neural network that can estimate the consequence of the non-simulated scenarios. Combining the simulation results and the ability of a deep neural network, Deep-SAILS can accurately identify the success and failure of the scenarios with the minimized number of physical model runs. The result of the case study for a small break LOCA shows that Deep-SAILS classified the success and failure of 10,143 scenarios while simulating only 11.13 % of the scenarios.

References

[1] J. Reynolds, Z. Jankovsky, K. G. Metzroth, and M. Darling, "*ADAPT*," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2018.

[2] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, R. Kinoshita, and A. Naviglio, "*RAVEN and dynamic probabilistic risk assessment: Software overview*," in Proceedings of ESREL European Safety and Reliability Conference, 2014.

[3] A. Alfonsi *et al.*, "*Raven theory manual*," Idaho National Lab.(INL), Idaho Falls, ID (United States), 2016.

[4] S. Baek, G. Heo, T. Kim, and J. Kim, "*Introduction to DICE (Dynamic Integrated Consequence Evaluation) toolbox for checking coverability of operational procedures in NPPs*," in 29th European Safety and Reliability Conference, ESREL 2019, 2020: Research Publishing Services, pp. 1588-1593.

[5] G. Heo, S. Baek, D. Kwon, H. Kim, and J. Park, "*Recent research towards integrated deterministic-probabilistic safety assessment in Korea*," Nuclear Engineering and Technology, 2021/05/24/ 2021, doi: https://doi.org/10.1016/j.net.2021.05.015.

[6] B. Echard, N. Gayton, and M. Lemaire, "*AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation*," (in English), Struct Saf, vol. 33, no. 2, pp. 145-154, 2011, doi: 10.1016/j.strusafe.2011.01.002.

[7] L. Puppo, N. Pedroni, A. Bersano, F. Di Maio, C. Bertani, and E. Zio, "*Failure identification in a nuclear passive safety system by Monte Carlo simulation with adaptive Kriging*," (in English), Nucl Eng Des, vol. 380, Aug 15 2021, doi: ARTN 111308
10.1016/j.nucengdes.2021.111308.

[8] L. Puppo, N. Pedroni, F. D. Maio, A. Bersano, C. Bertani, and E. Zio, "*A Framework based on Finite Mixture Models and Adaptive Kriging for Characterizing Non-Smooth and Multimodal Failure Regions in a Nuclear Passive Safety System*," Reliability Engineering & System Safety, vol. 216, p. 107963, 2021/12/01/ 2021, doi: https://doi.org/10.1016/j.ress.2021.107963.

[9] P. Turati, A. Cammi, S. Lorenzi, N. Pedroni, and E. Zio, "*Adaptive simulation for failure identification in the advanced lead fast reactor European demonstrator*," Progress in Nuclear Energy, vol. 103, pp. 176-190, 2018.

[10] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "*When Gaussian process meets big data: A review of scalable GPs*," IEEE transactions on neural networks and learning systems, vol. 31, no. 11, pp. 4405-4423, 2020.

[11] J. Bae and S. J. Lee, "*Limit Surface Search Algorithm With Artificial Neural Network and Monte Carlo Dropout Uncertainty Quantification*," 2021: PSA.

[12] J. Bae and S. J. Lee, "*Uncertainty-aware Limit Surface Search Algorithm using Deep Neural Network*," 2021: ANS.

[13] J. Bae, J. W. Park, and S. J. Lee, "*Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment*," Applied Soft Computing, vol. 124, p. 109007, 2022/07/01/ 2022, doi: https://doi.org/10.1016/j.asoc.2022.109007.

[14] Y. Gal and Z. Ghahramani, "*Dropout as a bayesian approximation: Representing model uncertainty in deep learning*," in international conference on machine learning, 2016: PMLR, pp. 1050-1059.

[15] J. B. J. W. P. S. J. Lee, "*Adaptive Sampling of Dynamic Scenarios Close to the Limit Surface using Deep Neural Network and Monte Carlo Dropout*," presented at the Korean Nuclear Society Autumn Meeting, 2021.

[16] C. Parisi, A. Alfonsi, D. Mandelli, and C. Rabiti, "*AUTOMATIC LIMIT SURFACE SEARCH FOR PWR TRANSIENTS BY RELAP5-3D/RAVEN CODES*," Idaho National Lab.(INL), Idaho Falls, ID (United States), 2018.

[17] J. Bae, G. Kim, and S. J. Lee, "*Real-time prediction of nuclear power plant parameter trends following operator actions*," Expert Systems with Applications, vol. 186, p. 115848, 2021/12/30/ 2021, doi: https://doi.org/10.1016/j.eswa.2021.115848.

[18] J. H. Shin, J. M. Kim, and S. J. Lee, "*Abnormal state diagnosis model tolerant to noise in plant data*," Nuclear Engineering and Technology, vol. 53, no. 4, pp. 1181-1188, 2021/04/01/ 2021, doi: https://doi.org/10.1016/j.net.2020.09.025.

[19] J. Choi and S. J. Lee, "*A Sensor Fault-Tolerant Accident Diagnosis System*," Sensors, vol. 20, no. 20, p. 5839, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/20/5839.
[20] J. Gawlikowski *et al.*, "*A Survey of Uncertainty in Deep Neural Networks*," arXiv preprint arXiv:2107.03342, 2021.
[21] Y. Gal, "*Uncertainty in deep learning*," Doctor of Philosophy, University of Cambridge, 2016.
[22] J. W. Park and S. J. Lee, "*Simulation optimization framework for dynamic probabilistic safety assessment*," Reliability Engineering & System Safety, vol. 220, p. 108316, 2022/04/01/ 2022, doi: https://doi.org/10.1016/j.ress.2021.108316.