Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 26-27, 2023

Postprocessing of Multi-dimensional Limit Surfaces Searched by a Deep Learning-Based Algorithm

Junyong Bae, Yunyeong Heo, Wooseok Jo, 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; yyheo0207@unist.ac.kr; cws5528@unist.ac.kr; sjlee420@unist.ac.kr

***Keywords: Dynamic PSA, Deep learning, Limit surface, Pareto frontiers**

1. Introduction

Probabilistic safety assessment (PSA) for nuclear power plants (NPPs) calculates risk by considering the interactions among diverse components and human responses, employing conservative assumptions regarding these factors. Due to the potential for these conservative assumptions to mask latent risks, supplementary methodologies have been suggested to integrate the dynamic behaviors of these factors. Nevertheless, these methodologies might augment the number of representative scenarios requiring simulation through computationally expensive physical models, such as thermal-hydraulic system codes. As a result, one of the fundamental practical challenges in dynamic PSA pertains to reducing computational burdens associated with simulating extensive scenarios.

In addressing this challenge, the author's previous research introduced a Deep learning-based Searching Algorithm for Informative Limit Surface/States/ Scenarios (Deep-SAILS) [1]. As depicted in Fig 1., the limit surface (LS) is a boundary between the regions of success and failure scenarios. Since the success or failure of any arbitrary scenario can be reasonably inferred using LS, pinpointing its location can be enlightening. Hence, this algorithm aims to identify LS while minimizing the number of simulations. Because it is hard to be explicitly located, this algorithm takes an alternative approach that is constraining LS through intensive simulations of scenarios proximate to the LS.

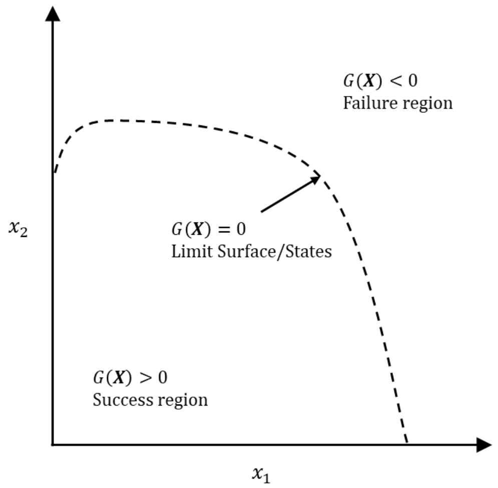

Fig. 1. Limit surface example [1]. x_1 and x_2 represent dynamic factors and $G(X)$ is a limit state function.

However, comprehending the LS with more than three dynamic factors can be intricate. One conceivable approach to grasp this multi-dimensional LS is to generate a cross-section; nevertheless, this method necessitates making static assumptions about certain dynamic factors.

To resolve this issue, this study proposes a postprocessing algorithm for the high-dimensional (i.e., more than three dynamic factors) LS searched by Deep-SAILS. This algorithm identifies a success box (i.e., hyperrectangle) that can cover a substantial portion of the successful scenarios, and then transforms the identified box into a decision tree. Through this algorithm, an intuitive representation of the high-dimensional success region bounded by LS can be provided.

This paper comprises five sections. Section 2 offers a concise overview of Deep-SAILS. The proposed postprocessing algorithm is illustrated in Section 3. Case study results for both the general case and scenarios involving distributed dynamic factors are presented in Section 4. Section 5 provides the concluding remarks for this paper.

2. Deep-SAILS

Deep-SAILS is an iterative process of locating the LS using the metamodel, as illustrated in Fig. 2. The algorithm starts by simulating extreme scenarios with the highest and lowest dynamic factor values. Then, in the next step, a deep learning metamodel is trained using these simulation results. After that, the algorithm picks the scenarios to be simulated. It is conducted by first identifying the suspected scenarios based on the predicted result and predictive uncertainty of each scenario and second randomly sampling the scenarios among the suspected ones. When most of the scenarios have already been simulated, then the algorithm wraps up and stops. If not, the algorithm requests the simulation of the sampled ones. For more detailed information about the algorithm, please refer to [1].

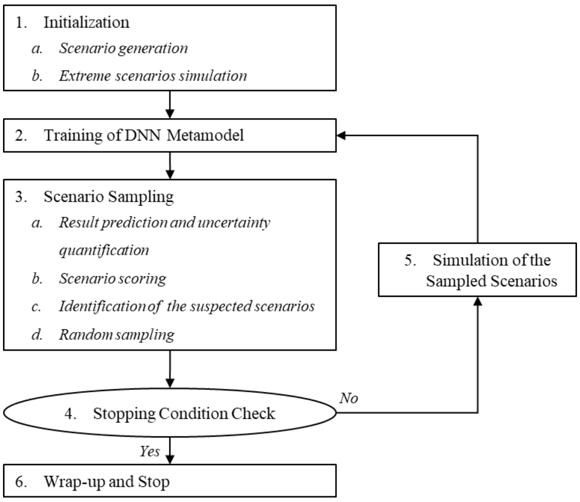

Fig. 2. Deep-SAILS flow chart [1]. Steps 2 – 4 are iterated until the stopping condition is satisfied.

Figure 3 provides an illustrative example of the LS located by the Deep-SAILS. The green and red dots correspond to simulated scenarios, and the background color indicates the estimated peak cladding temperature (PCT) predicted by the trained deep-learning model. The area meeting the core damage threshold (i.e., 1478K) is depicted as white. As shown in this figure, LS searching of Deep-SAILS outcomes simulation results of scenarios that were sampled, primarily near the LS, and the deep-learning model for non-sampled scenarios.

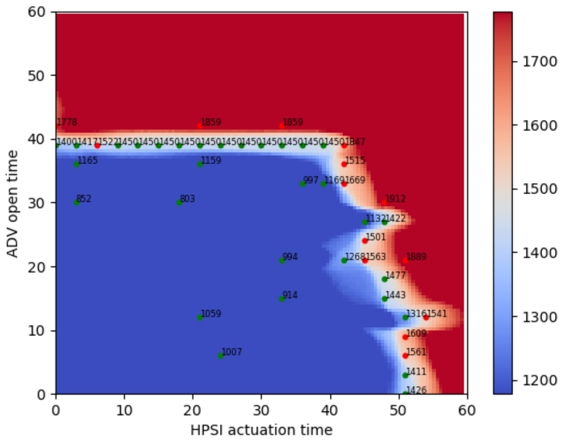

Fig. 3. LS searching result example [1].

3. Postprocessing of Limit Surface Searching Results

The objective of the postprocessing algorithm for Deep-SAILS is to identify the frontier scenario that defines the optimal success box (i.e., hyperrectangle), which contains success scenarios in the box as many as possible. In Fig. 4, frontier scenarios are depicted as blue dots and the identified scenarios as red dots, and the box configured by the identified one is shaded in red.

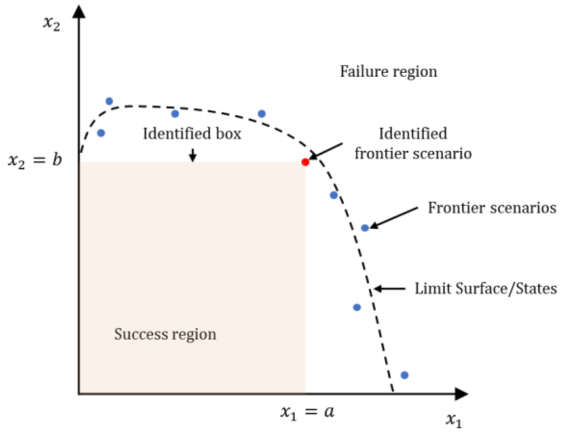

Fig. 4. Concept of frontier scenarios and success box.

As shown in the below Fig. 5, the success box can be easily converted into an easy-to-understand decision tree. This decision tree is a conservative yet intuitive representation of the LS, particularly for multi-dimensional cases.

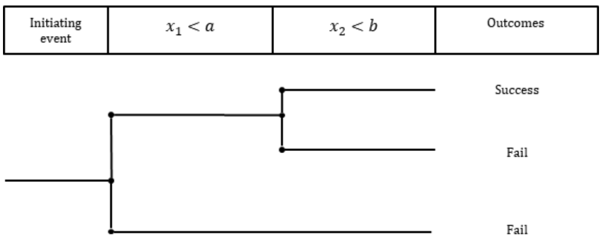

Fig. 5. Decision tree converted from the success box.

Similar research has been conducted by Park et. al [2, 3]. This research locates the LS by populating the success regions with multiple 'green' (i.e., success) boxes [3] and converts the green boxes into an event tree [2]. However, this research considers hypercube only. As a result, the boxes were not optimized to incorporate as many success scenarios as possible.

In contrast to the previous research, this study identifies a single box that encompasses as many success scenarios as possible. The algorithm comprises four major steps, as depicted in Figure 6, and is outlined below.

Step 1. *Initialization.* The initial stage involves identifying frontier scenarios through LS search outcomes. Frontier scenarios are those scoring a U-learning function [4] values lower than *D*, a suspicion range (a hyperparameter of Deep-SAILS) [1]. Additionally, to evaluate outcomes for non-simulated scenarios, a deep learning metamodel is employed.

Step 2. *Sorting by Hyperrectangle Size.* In this stage, the size of the success box formed by all frontier scenarios is computed. If the frontier is determined by k^{th} and l^{th} values of dynamic factors originating from success, the size is given by $k \times l$. Subsequently, the frontiers are organized based on their size.

Step 3. *Integrity Check.* The following step involves examining the surfaces of the success box, starting with the one having the largest size. If the surfaces contain failure scenarios, then the algorithm proceeds to scenarios with the subsequent largest size.

Step 4. *Decision Tree Conversion.* If the integrity of the success box is confirmed, a decision tree is constructed based on dynamic factor values of the corresponding frontier scenario.

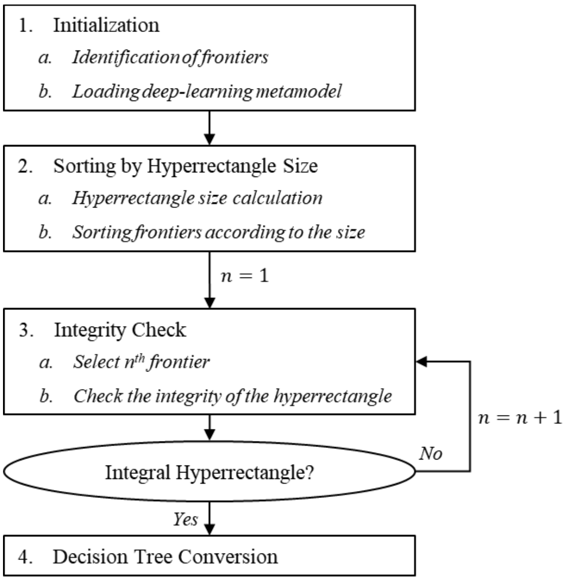

Fig. 6. Flowchart of the postprocessing algorithm.

4. Case Study

To validate the proposed algorithm, a case study involving arbitrary PCT data was conducted. Even though the algorithm is for the LS with more than three dynamic factors, we only assumed two dynamic factors for this case study to give illustrative examples. The dynamic factors, x_1 and x_2, are varying within the range of 0.01 to 0.99 with an increment of 0.02. As a result, a total of 2,500 scenarios were formulated. The PCTs for these scenarios were determined using Equation 1. The failure criterion was 1478 K.

$$(1) \qquad PCT(x_1, x_2) = 700(x_1^2 + x_2^2) + 700$$

Figure 7 shows the outcomes obtained from Deep-SAILS and the postprocessing algorithm. Similar to Fig. 3, the dots and background denote the simulated scenarios and PCT predictions by the deep-learning model, respectively. Deep-SAILS precisely pinpointed the LS and intensively simulated the scenarios in proximity to it. Additionally, the postprocessing algorithm successfully identified the success box containing the success scenarios as many as possible. The frontier scenario corresponding to this box is where $x_1 = 0.75$ and $x_2 = 0.73$. Consequently, the decision tree that can represent the LS is depicted in Fig. 8.

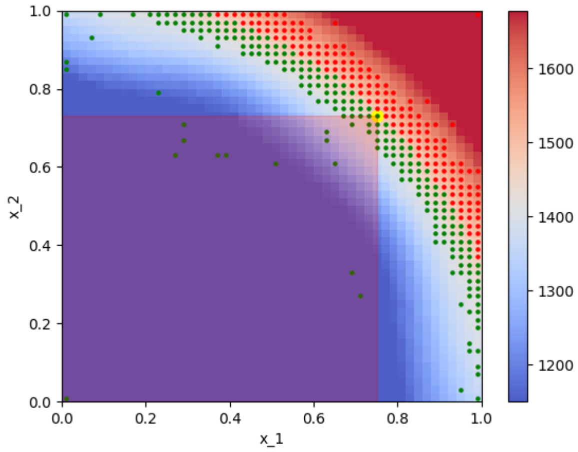

Fig. 7. Deep-SAILS results and optimal box (red-shaded area) with the identified frontier (yellow dot).

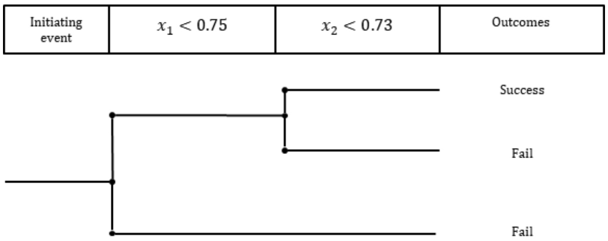

Fig. 8. Decision tree for the hyperrectangle in Fig. 7.

In some cases, a dynamic factor may adhere to a certain distribution, necessitating the sampling of axes according to this distribution. In such cases, the optimal success box should be adjusted, as depicted in Figure 9.

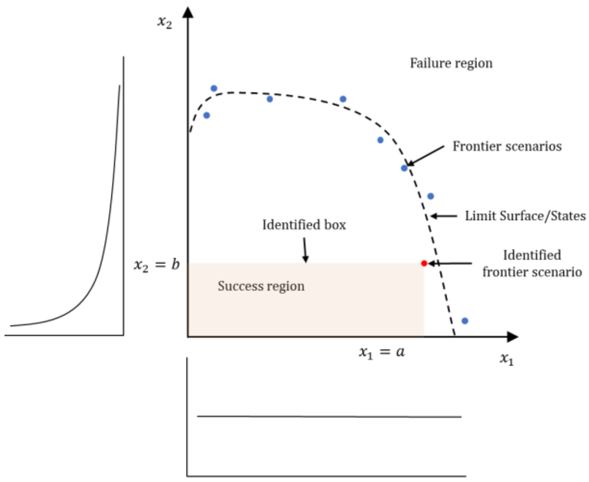

Fig. 9. Optimal frontier scenario and success box when dynamic factor follows a distribution.

To examine the behavior of the postprocessing algorithm when certain axes are sampled according to a distribution, we considered the scenarios where x_1

follows an exponential distribution [i.e., $x_1 \sim Exponential(3.0)$]. As a result, it was observed that the algorithm adjusted its initial estimation to $x_1 = 0.57$ and $x_2 = 0.87$.

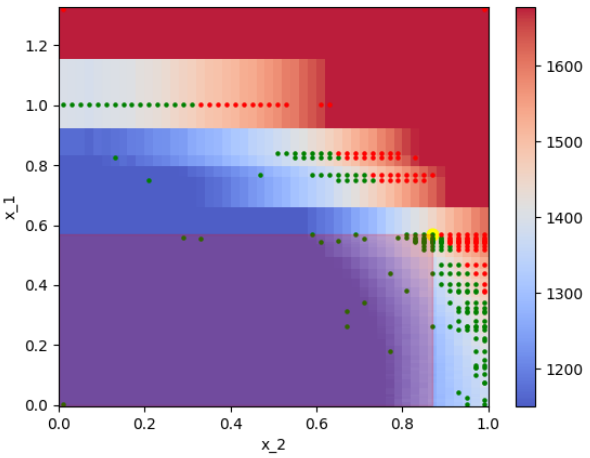

Fig. 10. Deep-SAILS results and optimal box with the identified frontier when $x_1 \sim Exponential(3.0)$.

Even though previous results show that the algorithm can optimize the success box to contains as many success scenarios as possible, the underlying limitation of the proposed postprocessing algorithm is that a single success box can remain a significant number of other success scenarios. Since this single-box algorithm may be too conservative and could mask the LS searching result, further study will be conducted to refine the algorithm to determine the optimal number of boxes that can cover most success scenarios This approach aligns with the study conducted by Park et al. [2, 3].

5. Conclusion

Localized scenarios (LSs) can offer valuable insights. However, understanding LSs becomes challenging in multi-dimensional contexts. To tackle this issue, this study proposes a postprocessing algorithm designed for LSs identified by Deep-SAILS. This algorithm identifies the optimal success box (hyperrectangle) that accommodates a maximum number of success scenarios and transforms it into a decision tree.

To verify the algorithm's effectiveness, case studies were carried out using both arbitrary data and the data with the dynamic factor following a distribution. Notably, these case studies were limited to low-dimensional scenarios, serving to demonstrate the algorithm's functionality. Therefore, future research will address high-dimensional scenarios involving more than three dynamic factors, along with plant accident scenarios and simulation through the system codes. Additionally, the process to determine the optimal number of boxes will be developed and integrated into the proposed algorithm.

REFERENCES

[1] 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, pp. 109007, 2022.

[2] J. W. Park, and S. J. Lee, Dynamic Event Tree Construction of Small LOCA based on Simulation Optimization Framework, Transactions of the Korean Nuclear Society Autumn Meeting, 2021

[3] J. W. Park, and S. J. Lee, Simulation optimization framework for dynamic probabilistic safety assessment," Reliability Engineering & System Safety, Vol. 220, pp. 108316, 2022.

[4] B. Echard, N. Gayton, and M. Lemaire, AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation, Structural Safety, Vol. 33, no. 2, pp. 145-154, 2011.