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

22. 06. 27
Junyong Bae

This presentation includes figures and tables 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
Introduction
Introduction
Practical problem of DPRA

**Practical challenge of simulation-based DPRA**

- **An enormous number of scenarios**
  - Considering dynamic behavior → A scenario in a static PRA is divided into several scenarios.
  - Each scenarios should be analyzed by TH code runs. → High computational cost

[1]

[1] ADAPT – Dynamic Event Tree Generation and Analysis
Structural reliability

- **Monte Carlo Simulation (MCS)**
  - Stochastic sampling of parameters
  - $P_f = (\# \text{ of failure scenarios})/(\# \text{ of total scenarios})$

- If a system failure is a rare event,
  - An enormous number of scenarios are required.
    - E.g., $P_f = 1 \times 10^{-4} \rightarrow At \ least \ 10,000 \ scenarios$

- Each scenario should be analyzed by complex computer codes. $\rightarrow$ High computational cost

- To tackle this problem, **surrogate models** are widely used.

\[ P_f = \frac{\text{The number of failure scenarios}}{\text{The number of total scenarios}} \]
An Active learning reliability method combining Kriging and MCS (AK-MCS)

- MCS without evaluating the whole scenario population.
- Success/failure of each scenario is predicted by a surrogate model based on a few simulated scenarios.

- Surrogate model: A kriging model (i.e., Gaussian process regression)
- A few simulated scenarios
  - AK-MCS iteratively samples and simulates the scenarios close to a limit surface.
    - Surrogate model can locate a limit surface.

[Introduction]

**Specialty of AK-MCS**

- Predictive uncertainty
  - Gaussian process regression: prediction + predictive uncertainty
  - The scenarios sampling efficiency can be improved by prioritizing the scenarios with not only closeness to the limit surface **but also high predictive uncertainty.**
    - Meticulous searching of limit surface
Introduction
AK-MCS for dynamic PRA

AK-MCS for NPPs
- Failure identification of
  ✓ Lead Fast reactor
  ✓ Passive safety system

AK-MCS for dynamic PRA
- It can minimize the number of simulations by locating a decision boundary.
- However, **different surrogate model should be employed**.
  ✓ Scenarios of dynamic PRA has their probability (↔ Monte Carlo simulation).
  ✓ Limit surface with more than thousands of adjacent scenarios
    - Gaussian process has a cubic time complexity $O(n^3)$ where $n$ is the size of the training data sets.

Novel algorithm for dynamic PRA

- Scenarios of dynamic PRA has their probability (↔ Monte Carlo simulation).
- Therefore, limit surface should be meticulously located by adjacent scenarios.
  ✓ The number of adjacent scenarios can be more than thousands.

- Novel algorithm should be able to address more than thousands scenarios
  ✓ Deep-learning model
  - Novel algorithm needs to keep the advantages of AK-MCS (i.e., predictive uncertainty).
    ✓ Monte Carlo dropout (MC dropout) and U-learning function

- Deep-SAILS
  ✓ Deep-learning based Searching Algorithm of Informative Limit Surface/Scenarios/States
02 Deep-SAILS
Informative scenarios

• The scenarios where the consequence is success or failure by a narrow margin.
  ✓ locating the limit surface/states
  ✓ Reasonable assumption about success or failure of remaining scenarios

Example

✓ LB LOCA
  ▪ x-axis: ESFAS delayed time
  ▪ y-axis: SI
  ▪ 1478K is success criterion.

✓ The scenarios in the yellow shaded area
Deep-SAILS

• Deep-learning based Searching Algorithm of Informative Limit Surface/Scenarios/States
  1. Simulation of informative (i.e., close to the limit surface) scenarios
  2. Consequence prediction for remaining scenarios with surrogate model
     → Identification of scenario success/failure with minimized simulations.

• Iterative algorithm
  1) Estimates limit surface
     ✓ Predicts a critical parameter (e.g., PCT) of whole scenarios.
     ✓ Deep-learning model
  2) Samples the scenarios close to the limit surface
     ✓ Consider closeness and prediction uncertainty together
        ✓ Monte Carlo dropout (MC dropout) and U-learning function
  3) Simulates the sampled scenarios, updates the deep-learning model, and locates limit surface

Deep-SAILS across a sea of dynamic scenarios...
Detailed Algorithm

1. Initialization
   ✓ Generation of a population of scenarios
   ✓ Preferential simulation of extreme scenarios
     ▪ Extreme scenarios are the scenarios configured by the maximum and minimum values of each parameter.

2. **Deep-learning model** training

3. **Scenarios sampling** (details in following slides)

4. Stopping condition
   ✓ The proportion of already simulated scenarios out of the sampled one.

5. Simulation of the sampled scenarios
Consequence prediction and predictive uncertainty quantification

- **Monte Carlo dropout**
  - Multiple inferences with different dropout configuration.
    - **Means and variance** can be interpreted as the **prediction and uncertainty**, respectively.

- Deep-SAILS predicts consequences of each scenario multiple times with a random dropout configuration.
  - Acceleration through efficient program and high-performance devices is necessary.
    - (Case study utilized TensorFlow 2.7 with CUDA, RTX3080 GPU, and I7-10700K processor)
Scenario Sampling with U-learning function

- Deep-SAILS scores each scenario by the **U-learning function** [1].

\[
U(X_i) = \frac{|\hat{y}_i - a|}{\sigma_{\hat{y}_i}} = \frac{|G(X_i)|}{\sigma_{\hat{y}_i}}
\]

✓ This function gives a lower score for the scenarios where the estimated consequence...
  - are closer to the failure condition (i.e., the numerator)
  - have higher uncertainty (i.e., the denominator)

- The suspected scenarios: \(U(X) < D\)
  ✓ \(D\) is the range of suspicion and critical algorithm hyperparameter

- Random sampling among the suspected scenarios
  ✓ Exploration

Case study
**Case study**

**TH system model**

**TH system**
- Zion NPP, WH-4 loop PWR, 1000MWe
- Assuming only SITs, LPSI, HPSI, and the SHRS via ADVs
## Metric
- Classification error rate

<table>
<thead>
<tr>
<th>True result</th>
<th>Predicted result</th>
<th>Simulation result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success $(a)$</td>
<td>$a\hat{a}$</td>
<td>Success $(A)$</td>
</tr>
<tr>
<td>Failure $(b)$</td>
<td>$b\hat{a}$</td>
<td>Failure $(B)$</td>
</tr>
</tbody>
</table>

\[
\text{Classification error percentage} = \frac{a\hat{b} + b\hat{a}}{a\hat{a} + a\hat{b} + b\hat{a} + b\hat{b} + aA + bB} \times 100
\]
# Case study (1): SB-LOCA

- 10,143 scenarios

✓ HPSI performance: **pump performance % = (Conservatively) flow rates %**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Uncertain domain</th>
<th>Discretization</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPSI actuation time</td>
<td>min</td>
<td>(0, 60)</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60)</td>
<td></td>
</tr>
<tr>
<td>ADV open time</td>
<td>min</td>
<td>(0, 60)</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60)</td>
<td></td>
</tr>
<tr>
<td>HPSI performance</td>
<td>%</td>
<td>(0, 100)</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(100, 92, 88, 83, 79, 75, 71, 67, 63, 58, 54, 50, 46, 42, 38, 33, 29, 25, 21, 17, 13, 8, 0)</td>
<td></td>
</tr>
</tbody>
</table>
Case study (1) : SB-LOCA

- 10,143 scenarios

- When $D$ (Range of suspicion) = 2.0
  ✓ Scenarios : 10,143
  ✓ Simulation : 1,129
  ✓ non-CD to non-CD : 3,776
  ✓ non-CD to CD : 1
  ✓ CD to CD : 6,365
  ✓ CD to non-CD : 1

- Error rate : 0.02%
  ✓ Simulation : 9.36%

Limit surface estimated by the deep learning model for each iteration
(Assume HPSI performance = 75%)
(14,400 predictions per frame)
Case study (1) : SB-LOCA

- 10,143 scenarios

- When D (Range of suspicion) = 2.0
  - Scenarios : 10,143
  - Simulation : 949
  - non-CD to non-CD : 3,776
  - non-CD to CD : 1
  - CD to CD : 6,365
  - CD to non-CD : 1

- Accuracy : 99.98%
- Simulation : 9.36%

Limit surface estimated by the deep learning model for each iteration

(Assume HPSI performance = 25%)
(14,400 predictions per frame)
**Case study (2) : LB-LOCA**

- 40,250 scenarios
- ✓ SIT performance: partial opening of cascading two valves.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Uncertain domain</th>
<th>Discretization</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIT-1 performance</td>
<td>%</td>
<td>(0, 100)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 25, 50, 75, 100)</td>
</tr>
<tr>
<td>SIT-2 performances</td>
<td>%</td>
<td>(0, 100)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 25, 50, 75, 100)</td>
</tr>
<tr>
<td>SIT-3 performances</td>
<td>%</td>
<td>(0, 100)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 25, 50, 75, 100)</td>
</tr>
<tr>
<td>ESFAS delayed time</td>
<td>s</td>
<td>(0, 400)</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 400)</td>
</tr>
<tr>
<td>LPSI performance</td>
<td>%</td>
<td>(0, 100)</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(100, 92, 88, 83, 79, 75, 71, 67, 63, 58, 54, 50, 46, 42, 38, 33, 25, 21, 17, 13, 8, 0)</td>
</tr>
</tbody>
</table>
Case study (2) : LB-LOCA

- 40,250 scenarios

- When D (Range of suspicion) = 2.0
  - Scenarios : 40,250
  - Simulation : 6,140
  - non-CD to non-CD : 17,725
  - non-CD to CD : 31
  - CD to CD : 22,488
  - CD to non-CD : 6

- Accuracy : 99.91%
- Simulation : 15.25%

Limit surface estimated by the deep learning model for each iteration

(Assume SIT performances = 50%)
(14,400 predictions per frame)
Case study (2) : LB-LOCA

- 40,250 scenarios

- Trade off relationship between accuracy and simulation according to D (range of suspicion)

<table>
<thead>
<tr>
<th></th>
<th>$D = 0.5$</th>
<th>$D = 2.0$</th>
<th>$D = 5.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenarios</td>
<td>40,250</td>
<td>40,250</td>
<td>40,250</td>
</tr>
<tr>
<td>Simulation</td>
<td>4,412</td>
<td>6,140</td>
<td>11,293</td>
</tr>
<tr>
<td>non-CD to non-CD</td>
<td>17,672</td>
<td>17,725</td>
<td>17,749</td>
</tr>
<tr>
<td>non-CD to CD</td>
<td>84</td>
<td>31</td>
<td>7</td>
</tr>
<tr>
<td>CD to CD</td>
<td>22,462</td>
<td>22,488</td>
<td>22,494</td>
</tr>
<tr>
<td>CD to non CD</td>
<td>32</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Accuracy</td>
<td><strong>99.71%</strong></td>
<td><strong>99.91%</strong></td>
<td><strong>99.98%</strong></td>
</tr>
<tr>
<td>Simulation (%)</td>
<td><strong>10.96%</strong></td>
<td><strong>15.25%</strong></td>
<td><strong>28.06%</strong></td>
</tr>
</tbody>
</table>
Case study (2) : LB-LOCA

- 40,250 scenarios

- Trade off relationship between accuracy and simulation according to D (range of suspicion)

![Graphs showing the relationship between LPSI performance and ESFAS delayed time for different values of D (0.5, 2.0, and 5.0).]
Case study (3) : Modified Rastrign function

- 1,771,561 scenarios

- \( f(x) = 60 + \sum_{i=1}^{6}[x_i^2 - 10\cos(2\pi x_i)] \)
  - \( x_i \in \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\} \)
  - 6 factors with 11 performances = 1,771,561

- Assume \( f(x) > 90 \) as failure criteria
  - 158,640 fail scenarios
  - Failure rate (assume constant sampling distribution) = 0.0899
Case study (3) : Modified Rastrign function

- 1,771,561 scenarios

\[
x_3, x_4, x_5, x_6 = 0, 0, 0, 0
\]

\[
x_3, x_4, x_5, x_6 = 0, 0, 5, 5
\]

\[
x_3, x_4, x_5, x_6 = 5, 5, 5, 5
\]
Case study (3) : Modified Rastrign function

- 1,771,561 scenarios

- When D (Range of suspicion) = 2.0
  ✓ Scenarios : 1,771,561
  ✓ Simulation : 60,525

  ✓ non-CD to non-CD : 1,605,241
  ✓ non-CD to CD : 0
  ✓ CD to CD : 158,640
  ✓ CD to non-CD : 6

  ✓ Accuracy : 100.00%
  ✓ Simulation : 3.431%

Note: Division by zero problem for U-learning function exists
Conclusion
• This research introduced a guided simulation algorithm of a dynamic PRA, named **Deep-SAILS**.
• To overcome the limitation of previous including AK-MCS, the algorithm employed a **deep-learning model** 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 learning model that can estimate the consequence of the non-simulated scenarios**.
• Consequently, Deep-SAILS can accurately identify the success and failure of the scenarios with the **minimized number of physical model runs**.
• Case study result shows the effectiveness of Deep-SAILS


• Detailed information can be found in following articles:
  ✓ Junyong Bae et. al., *Limit surface/states searching algorithm with a deep neural network and Monte Carlo dropout for nuclear power plant safety assessment* (2022)
  ✓ Jong Woo Park et. al., *Simulation optimization framework for dynamic probabilistic safety assessment* (2022)
Thank you

Any Questions?

SAILS across a sea of dynamic scenarios...
Case study (1) : SB-LOCA

- 10,143 scenarios

- Comparison with other method
  ✓ Random sampling (Random)
  ✓ Support vector machine (SVM)
    ▪ ~ Adaptive sampler of RAVEN
Case study (3) : Modified Rastrign function

- 1,771,561 scenarios
  - Number of simulated scenarios for each iteration
  - Normalized failure scenarios (Estimated number / Real number)

Note: Division by zero problem for U-learning function exists