Comparative Analysis of Surrogate Models with Recurrent Neural Network and Interpolation Method

Donghee Choi[1], Man Cheol Kim[2]

[1] *Department of Energy Systems Engineering, Chung-Ang University: 84 Heukseokro, Dongjak-gu, Seoul, Korea, 06974, donghe0128@cau.ac.kr*

[2] *Department of Energy Systems Engineering, Chung-Ang University: 84 Heukseokro, Dongjak-gu, Seoul, Korea, 06974, charleskim@cau.ac.kr*

ABSTRACT

Probabilistic safety assessment (PSA) is a key method for quantifying the frequency and consequences of various accident scenarios. Recently, there has been a growing demand for considering dynamic interactions between operator actions and thermal-hydraulic (TH) behavior over time, leading to active research on dynamic PSA. However, dynamic PSA requires generating scenarios in real time and repeatedly running TH codes for each scenario, resulting in significantly increased computation time and resource consumption. To accelerate this, several neural network-based surrogate models have been proposed, but issues such as long training times, out-of-bounds prediction risks, and difficulties in interpreting results remain. As an alternative to avoid these limitations, a simple interpolation-based surrogate model can be proposed. In this study, we propose a surrogate model constructed using multidimensional linear interpolation and compare it with a neural network surrogate model trained on the same dataset. The target scenarios include Loss of Feed Water accident scenarios involving operator response delays. Comparing the mean relative error, training time, and prediction time, the interpolation model achieved accuracy nearly identical to that of the neural network model, with millisecond prediction time without requiring a training process. In conclusion, selecting the interpolation model instead of long-term neural network training within the given computing resources and time can reduce requirements while maintaining the accuracy for TH analysis. This suggests a practical alternative that achieves a balance between efficiency and accuracy.

Keywords: Probabilistic safety assessment, Dynamic PSA, Surrogate model, Neural network, Interpolation

I. INTRODUCTION

Probabilistic safety assessment (PSA) has long been the standard method for stochastically estimating how often nuclear plant accidents might occur and how serious their outcomes could be. Conventional PSA uses logic-based event tree and fault tree models with simple success or failure branches. Several studies have said that this structure has trouble capturing the continuous, time-dependent interactions that arise from operator actions during an accident [1]. To address these dynamic interactions, dynamic PSA (DPSA) was introduced. DPSA aims to simultaneously generate accident scenarios and conduct thermal-hydraulic (TH) simulations in real time [2]. Running the TH calculations for each time step across thousands or tens of thousands of dynamic scenarios requires considerable computing and time resources. Researchers often simplify the timeline into a few representative time intervals or introduce bounding assumptions, yet simulation time still increases rapidly with scenario count.

Conventional PSA uses TH analysis to set success criteria, whereas DPSA bases every scenario evaluation on a TH simulation. When the branch count increases, each branch needs another TH analysis. The fine time steps required for high-resolution results drive computation and time resource use upward, so high-resolution DPSA remains impractical on ordinary hardware. To resolve this burden, many researchers have considered neural network (NN) surrogate models. Chae et al. proposed a lightweight DPSA framework that integrates a multi layer physics-informed neural network (PINN). After training, the model predicted primary system pressure and peak cladding temperature (PCT) in milliseconds, enabling real-time assessment of scenarios [3]. Baraldi et al. applied a PINN to a loss of heat sink transient, adding reactivity feedback terms to the loss function so that the surrogate preserved physical consistency, reducing run time [4]. Also, He et al. developed a deep learning reduced order model that mapping the nonlinear behavior of steam generator TH, offering an economical alternative for repeated simulations [5]. These studies show that using NN surrogates to accelerate TH analysis has become an active line of research.

However, all NN models share fundamental limits. They need a training process and must be retrained whenever plant conditions or scenarios change. The optimal network architecture often differs from one dataset to another, and a network trained on one accident family may not perform equally well on a different one. In practice, analysts may have to develop and tune separate networks for each scenario group. Moreover, neural outputs are often hard to interpret, and extra tuning is required to prevent unexpected outputs. A key issue is that we still cannot easily explain why the network produces a particular result or predict how it will behave beyond the training dataset range.

A more clear alternative is to interpolate a precomputed dataset. Because interpolation works as soon as a database exists, it provides almost instant predictions and remains within that dataset range. Bennett et al. applied sparse-grid interpolation to predict subchannel void fractions in a boiling water reactor with low error and millisecond response time [6]. Lee et al. used kriging interpolation to map the failure surface, reducing the required TH runs from thousands to just tens [7]. These cases insight that interpolation can be used as a fast and clear surrogate model. Building on this, the present study uses the same TH dataset to compare a NN based surrogate model with a multidimensional linear interpolation model in accidents: loss of feed water accidents (LOFW). Thereby offering quantitative insights into their relative performance.

II. METHODOLOGY

II.A. Scenario Definition and Dataset

Table I summarises the structured grid dataset used by both surrogate models. For the accident family, the grid of input variables, the number of scenarios, the output vectors for pressurizer pressure (PZR-P) and PCT, and the resulting dataset size (time points × scenarios). In total, the dataset comprises 20 scenarios and 5,000 time points each, resulting in a total of 100,000 data points.

In the LOFW accident family, the transient begins at t = 0s with a main feed water system trip and a failed auxiliary feed water system. With no secondary heat removal, primary pressure increases until it reaches the pressurizer safety valve (PSV) setpoint. Because the PSV is assumed to be stuck closed, the operator must switch to a feed and bleed operation manually. At that time, the operator opens the safety depressurisation system (SDS) after a specified delay $t_{sds} \in \{0,100,200,300s\}$. When the decreasing primary pressure reaches the injection threshold of high-pressure safety injection (HPSI), we assume that the HPSI does not start automatically. Instead, the operator manually starts the HPSI after an additional chosen delay $t_{hpsi} \in \{100,200,300,400,500s\}$.

Figure 1 shows the PCT curve for the 20 LOFW scenarios described above. All cases remain nearly flat until feed and bleed begin, once SDS opens and HPSI starts, the PCT drops by the chosen t_{sds} and t_{hpsi}. The monotonic, gradually declining nature of these curves supports the use of relatively shallow surrogate architectures. Figure 2 shows a sample of PZR-P curve for the 20 LOFW scenarios. The sharp pressure drop right after each SDS opening (red box), and the point where HPSI starts (blue box) can also to seen. The differences between the curves show how the four SDS delays and five HPSI delays change the timing of these operator actions.

TABLE I. Data Set

Accident family	Independent variables	No. of scenarios	Output vector (t, …)	Matrix size
LOFW	SDS opening delay (t_{sds}): 0~300s, ($\Delta t_{sds} = 100s$)	$4 \times 5 = 20$	$PZR\ P\ (t, t_{sds}, t_{hpsi})$ $PCT\ (t, t_{sds}, t_{hpsi})$	20×5000
	HPSI start delay (t_{hpsi}): 100~500s, ($\Delta t_{hpsi} = 100s$)			
	Run time (t): 5000s, ($\Delta t = 1s$)			

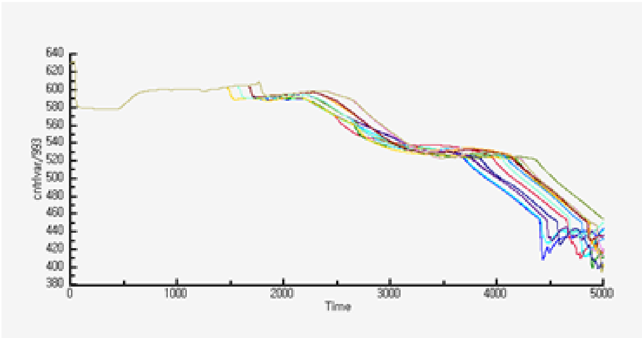

FIGURE 1. PCT of LOFW

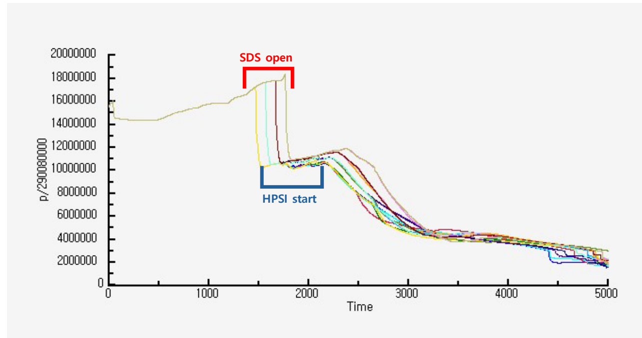

FIGURE 2. PZR-P of LOFW

II.B. Recurrent Neural Network Surrogate Model

The pressure and temperature parameters generated by the TH analysis are long and smooth time series, thus a recurrent neural network (RNN) surrogate is selected for the NN surrogate model. Among RNN, we selected the long short-term memory (LSTM) network because its gating mechanism preserves long range temporal information while mitigating the vanishing gradient problem. The modest data volume of 20 LOFW scenarios and the monotonic shape of the curves make a single LSTM layer an effective compromise. Tests with deeper layers showed no meaningful accuracy gain but higher training time and overfit risk. Table II lists the two constructions used. Both models use the same three input vector $X_t = \left[t, t_{sds}, t_{hpsi}\right]^T$. In the baseline model, the hidden layer contains 64 LSTM units and a dropout rate of 0.2 to prevent overfitting. For LOFW, a second model (LOFW-B) extends the layer to 128 units and the epoch from 50 to 2000 to probe the maximum accuracy with other hyperparameters remaining unchanged. A single fully connected node converts the LSTM output to one value, and the network is trained with a mean squared error loss. We use 90 % of the existing data for training, and the remaining 10 % is taken from the same dataset for validation. The latter acts as an online validation set. Throughout training, the MATLAB progress plots showed no noticeable overfitting, and the 0.2 dropout layer contributed to maintaining that stability even in the 2000 epoch LOFW-B run. Weights are tuned to minimize loss:

$$L = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\left(\hat{y}_{t,n} - y_{t,n}\right)^2, \tag{1}$$

where N is the number of training scenarios (90% of LOFW scenarios), T= 5000 is the number of time steps in each run. The $y_{t,n}$ is the true value of normalised PZR-P or PCT at time t in scenario n, taken from the target data set Y_{train}. On the other hand, $\hat{y}_{t,n}$is the network's predicted value at the same moment, computed from the input sequence X_{train}.

TABLE II. RNN Model

Model	Sequence input	Units	Dropout	Epochs	Batch	Optimiser	Learning rate
Baseline (LOFW-A)	3 feature (t, t_{sds}, t_{hpsi})	1 LSTM, 64 units	0.2	50	128	Adam	5×10^{-3}
Accuracy-focused (LOFW-B)	3 feature (t, t_{sds}, t_{hpsi})	1 LSTM, 128 units	0.2	2000	128	Adam	5×10^{-3}

II.C. Multidimensional Interpolation Model

The interpolation surrogate model directly uses the precomputed TH results stored on a structured three-dimensional grid. Therefore, no additional training or model tuning is required. With this grid, trilinear interpolation can estimate PCT or PZR-P at any arbitrary query point inside the domain. This approach ensures that predictions remain within physically validated boundaries and delivers instantaneous results. For the LOFW grid used here, the three axes are simulation time t, SDS opening delay t_{sds}, t_{hpsi}. At every grid node (t_i, t_j, t_k) each node holds the pre-calculated outputs for PCT and PZR-P. When an arbitrary query point $(t^*, {t_{sds}}^*, {t_{hpsi}}^*)$ is requested, the algorithm first identifies the cube that encloses that point, picks out the eight corner nodes, and determines the fractional distances (α, β, γ) of the query point relative to the cell's origin corner along all three axes. Those fractional distances serve as linear weights, the closer the query point is to a particular corner, the larger the share of that corner's value in the final output. By combining the eight stored node values in this way, the model delivers an immediate, smoothly varying estimate that is guaranteed to stay within the range of the original data. In Section III, we will validate the interpolation model and an LSTM surrogate model for the accident scenario, comparing accuracy and prediction time to confirm their performance.

III. CASE STUDY

In this section, we compare the performance of the LSTM surrogate model and the multidimensional interpolation surrogate model for the LOFW accident scenario. For the scenario, the structured dataset is first used to train the LSTM network. Subsequently, ten independent test cases are generated by Latin-Hypercube random sampling inside the same parameter bounds as the original domain (t_{sds}, t_{hpsi} for LOFW). For each test case, we run the TH analysis for the entire simulation time (5000s for LOFW) to obtain the simulation time series outputs of PZR-P and PCT. We then compare these series with the outputs produced by the LSTM model and the interpolation model.

The evaluation metrics include:

1. LSTM model's training and prediction time.
2. Interpolation model's prediction time.
3. Mean relative error, defined by:

$$RelErr(\%) = \frac{1}{T}\sum_{t=1}^{T}\frac{|\hat{y}(t) - y_{true}(t)|}{|y_{true}(t)|} \times 100$$

where T=5000s for simulation of LOFW, $\hat{y}(t)$ denotes the surrogate model's prediction at time t, and $y_{true}(t)$ is the corresponding TH computed results.

III.A. LOFW Families

For the LOFW, the LSTM was trained on the 20 cases grid shown in Table I, which covers the 5000s simulation and the two operator delay variables(t_{sds}, t_{hpsi}) in feed and bleed, and the interpolation model works on that same grid without any training. Ten independent query points ($t^*, t^*_{sds}, t^*_{hpsi}$) were then generated by Latin-Hypercube sampling inside this domain, and TH analysis provided the PCT and PZR-P time series for the test. Because LOFW shows a sharp pressure change once feed and bleed begin, two LSTM configurations were evaluated that the baseline network (LOFW-A) and the accuracy-focused network (LOFW-B) listed in Table II. Additionally, in the case of PZR-P, LOFW-A first achieved an in-domain validation mean relative error of 10.1%, while the accuracy-focused LOFW-B reduced this to 1.9%, confirming that no overfitting occurred before the Latin-Hypercube sampling test cases were assessed.

Figures 3 and 4 illustrate how closely both surrogate models follow the PCT curves. The blue interpolation lines and red LSTM lines overlap the black reference curves, confirming that both approaches capture the overall trend and timing of the LOFW scenarios. In LOFW-A, the baseline LSTM follows the general trend but lags behind the sudden drop that follows delayed injection, whereas the interpolation line almost overlaps with the actual curve. With LOFW-B, the accuracy-focused LSTM closes this gap, its trend almost coinciding with both the interpolation and the reference curves over the whole transient. Figures 5 and 6 compare the surrogate models with the PZR-P results. Like in the PCT case, the baseline LSTM (LOFW-A) under-predicts some sharp variations, while interpolation matches them closely in this scenario. With LOFW-B, the error falls to almost the same level as interpolation. Both surrogates predict PCT more accurately than PZR-P because the PCT curve is more monotonic.

Table III confirms the numerical trends. It confirms that interpolation returns its predictions almost immediately (prediction time ≈ 0.01s) with no training, whereas LOFW-A needs about 60s of training and 1s of prediction time. After 2000 training epochs, the accuracy-focused LSTM (LOFW-B) narrows this gap, but only after about 3000s of additional training time. The key difference remains computational speed. With comparable accuracy, interpolation offers advantages in computation time and implementation effort. While an accuracy-focused LSTM can close the gap, the extra accuracy comes at a high training cost. Although extending LSTM training even further could, in principle, improve its accuracy beyond interpolation, the extra computation time and resources may outweigh the benefit. We have to consider this trade-off. Invest more time and resources to get a slightly better result with NN, or use the simpler interpolation model with conservative assumptions, which is already quite accurate and saves both computational time and computing cost.

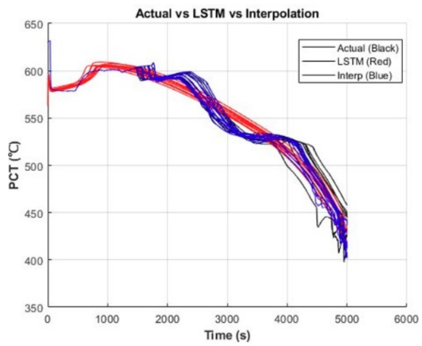

FIGURE 3. PCT Prediction of LOFW-A

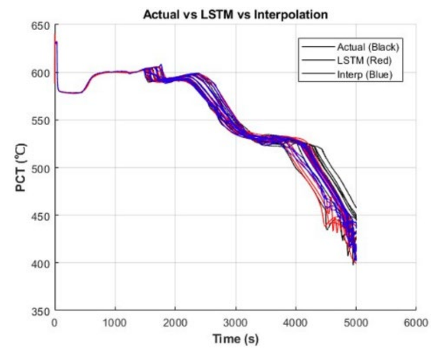

FIGURE 4. PCT Prediction of LOFW-B

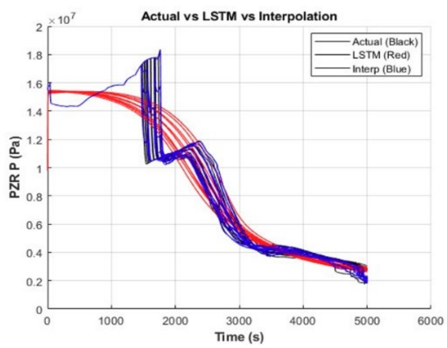

FIGURE 5. PZR-P Prediction of LOFW-A

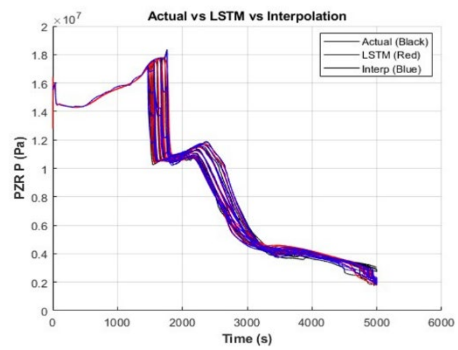

FIGURE 6. PZR-P Prediction of LOFW-B

TABLE III. LOFW Case

Model	Output	Mean relative error (%)	Training time (s)	Prediction time (s)
RNN (LSTM-A)	PCT	1.63	63.71	1.31
	PZR-P	9.66	54.63	0.57
RNN (LSTM-B)	PCT	0.78	2990.12	1.62
	PZR-P	2.94	2957.58	0.84
Interpolation	PCT	0.62	–	0.01
	PZR-P	2.54		

IV. CONCLUSION AND DISCUSSION

Conventional PSA simplifies accident sequences into binary success or failure branches and sets success criteria from a thermal-hydraulic (TH) analysis. DPSA, in contrast, subdivides these branches to consider operator actions and other dynamic interactions, requiring multiple TH analyses for each branch. On the other hand, DPSA subdivides the branches to reflect operator actions and dynamic interactions, and conducts multiple TH analyses for each branch. As a result, calculation time and resource consumption increase exponentially, and to reduce this, NN-based surrogate models are being proposed. NN is an alternative to reduce computational burden, but it introduces a new trade-off between accuracy improvement and extra training cost. They require a long learning time, can predict outside the data range, and are hard to interpret. If a simple interpolation model can deliver almost NN accuracy with less effort, it becomes an attractive alternative. To validate that, we built a simple three-dimensional trilinear-interpolation surrogate and compared it with an LSTM surrogate trained on the same data. We evaluated the mean relative error, training time, and prediction time for the LOFW scenario that includes operator delay variables. The interpolation model produced near instant predictions with almost the same accuracy as the basic LSTM, which required about 60s of training time. An accuracy-focused model reduced the error further but increased training time to roughly 3000s. The case study shows that even simple interpolation can achieve performance comparable to that of NN under the same dataset. Therefore, the choice between the two methods depends on the analysis objectives and available resources, making it a trade-off issue. Once the data exists, the interpolation model runs immediately, keeps every prediction inside that grid, and is easy to explain. The LSTM, in contrast, has to be retrained whenever the scenario family or data set changes. In conclusion, we confirmed that even a simple interpolation model can achieve performance comparable to the NN model within the scope of this study.

There are several limitations to this study. The LSTM model used in this study may not be the most optimized model for the dataset. Also, the comparison covered only LOFW accident families, so we need to compare more cases. Since a simple interpolation model was performed, trends and dependencies within the data were not accurately considered. Based on an understanding of how the variables in TH analysis move, it is expected that we are able to develop an interpolation method suitable for each situation.

ACKNOWLEDGMENTS

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Ministry of Trade, Industry and Energy (MOTIE) of Republic of Korea [grant number RS-2024-00398867].

REFERENCES

[1] Jong Woo Park, and Seung Jun Lee. "Simulation optimization framework for dynamic probabilistic safety assessment." Reliability Engineering & System Safety 220, 108316 (2022).
[2] Karanki, Durga Rao, and Vinh N. Dang. "Quantification of dynamic event trees–a comparison with event trees for MLOCA scenario." Reliability Engineering & System Safety 147, 19-31 (2016).
[3] Chae, Young Ho, et al. "Development of a data-driven simulation framework using physics-informed neural network." Annals of Nuclear Energy 189, 109840 (2023).
[4] Baraldi, Piero, et al. "Physics-Informed Neural Networks for the safety analysis of nuclear reactors." Progress in Nuclear Energy 185, 105745 (2025).
[5] He, Shaopeng, et al. "A deep-learning reduced-order model for thermal hydraulic characteristics rapid estimation of steam generators." International Journal of Heat and Mass Transfer 198, 123424 (2022).
[6] Bennett, Alexander, et al. "A surrogate model based on sparse grid interpolation for boiling water reactor subchannel void distribution." Annals of Nuclear Energy 131, 51-62 (2019).
[7] Lee, Yongjoon, and Gyunyoung Heo. "Sampling Based Method for Estimating Functional Failure Probability of Passive Safety Systems in Small Modular Reactors." Transactions of the Korean Nuclear Society Spring Meeting, (2025)