Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 22-23, 2025

Application and Limitations of Deep Learning for Predicting Severe Accident Progression

Joon Young Bae [a], JinHo Song [a], Sung Joong Kim[a, b]
[a]*Department of Nuclear Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, Korea*
[b]*Institute of Nano Science and Technology, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, Kore*
[*]*Corresponding author: sungjkim@hanyang.ac.kr*

***Keywords :** Deep learning, Physics-Informed Neural Networks, Severe accident codes

1. Introduction

Integrated severe accident analysis codes—such as CINEMA, MELCOR, MAAP, and ASTEC—have long been employed by industry, regulatory authorities, and the academic community for evaluating severe accidents in nuclear power plants (NPPs) [1]. These codes utilize complex numerical models to simulate detailed physical phenomena; however, the resulting computational burden often leads to prohibitively long simulation times, rendering them unsuitable for real-time decision-making.

Recent advances in data-driven techniques, particularly deep learning, have shown promise in expediting these prediction processes [2]. Deep learning, with its universal function approximation capability, can be trained on simulation outputs from severe accident codes and subsequently generate rapid predictions, facilitating near- or real-time assessment. Moreover, emerging research has explored deep reinforcement learning to automate and optimize accident mitigation strategies [3].

Although these developments, conventional deep learning models are frequently regarded as "black boxes," presenting challenges in terms of interpretability and regulatory acceptance [4]. Concerns persist regarding the extent to which data-driven methods can supplant—or complement—traditional physics-based analyses. Physics-Informed Neural Networks (PINNs) have been proposed to integrate physical constraints directly into neural network architectures [5], but practical limitations remain, such as difficulties in scaling to high-dimensional, multi-variable systems and sustaining real-time performance under substantial computational demands.

Even so, deep learning holds significant potential for expediting severe accident prediction. In this paper, we focus exclusively on revisiting the technical and practical constraints that arise when applying deep learning-based models derived from integrated accident analysis codes. Through this examination, we aim to elucidate the current limitations and suggest directions for addressing them in future work.

The remainder of this paper is organized as follows. Section 2 offers a brief review of the literature on deep learning applications in severe accident analysis. Section 3 introduces representative deep learning-based prediction models and outlines their methodologies. Section 4 provides an in-depth discussion of the technical and practical constraints these models face in real-world scenarios, including issues of scalability, interpretability, and computational efficiency. Finally, Section 5 concludes the paper by summarizing key findings and highlighting possible avenues for future research.

2. Methods and Results

2. Basic Concepts of Numerical Solvers, Deep Learning, and Physics-Informed Neural Networks

Figure 1 highlights the distinct yet complementary roles of traditional numerical solvers, purely data-driven deep learning (DNNs), and physics-informed neural networks (PINNs).

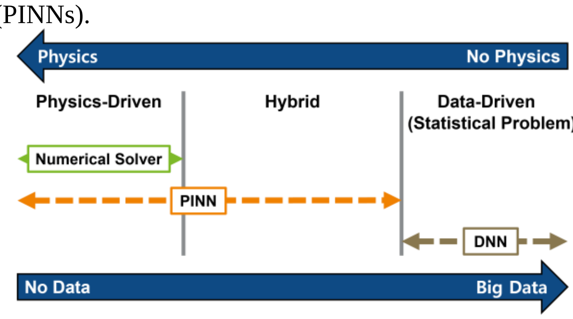


Figure 1 Conceptual diagram positioning the numerical solver, PINN, and DNN along the two axes: from purely physics-based approaches (no data) to purely data-driven approaches (big data), and from full physics to no physics.

Numerical solvers refer to algorithmic frameworks that approximate solutions to governing equations, such as ordinary or partial differential equations, by discretizing the continuous domain and iteratively computing approximate values of the solution variables.

Purely data-driven DNNs aim to learn functional relationships directly from large datasets, often without explicit consideration of underlying physical laws. Such models rely on the premise that, with sufficient data and model capacity, DNNs can approximate complex input–output mappings. However, in domains where data are scarce or expensive to obtain, purely data-driven approaches may overfit or generalize poorly, as they lack the inductive biases that physical constraints can provide.

PINNs explicitly integrate physical laws (e.g., partial differential equations) into the neural-network training process by penalizing deviations from these governing equations in the loss function. As a result, the model is constrained to respect conservation principles and boundary or initial conditions, even in data-scarce scenarios. This synergy between data-driven learning and first-principles knowledge improves model generalizability and interpretability, ensuring that physically meaningful solutions are learned.

3. Surrogate Modeling via Deep Learning: Process and Limitations of Purely Data-Driven Approaches

3.1. Process of Surrogate Modelling Using Conventional Deep Learning

The prediction of severe accident progression via a conventional deep learning-based, data-driven approach is accomplished through the process illustrated in Figure 1. In this process, surrogate models are developed that emulates the integrated severe accident analysis codes. Although various surrogate modelling approaches exist, in this paper the term "surrogate modelling" is used exclusively to refer to deep learning-based, data-driven methods [6].

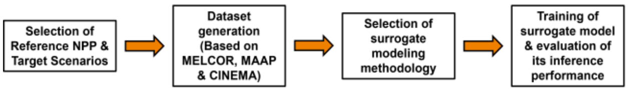


Figure 2 Overall research process for data-driven methodologies for surrogate modelling

In the process illustrated in Figure 2, the first step is the selection of a reference NPP and target scenarios. During this stage, a list of Engineered Safety Feature operations and failures, as well as mitigation strategies using Multi-barrier Accident Coping STrategy (MACST), is established. Following this, a dataset is constructed using the integrated severe accident analysis code—a step that is both computationally intensive and critical, as the quality of the surrogate model is directly dependent on the quality of the dataset. The primary objective of this phase is to compute scenarios by varying the sampling of the implementation timings for safety system operations, failure timings, and the implementation of mitigation strategies within the target scenarios. Subsequently, surrogate models are designed based on deep learning (artificial neural networks) and trained using the generated dataset. The evaluation of the trained surrogate model is based on the similarity between its predicted time series and the "true" values—assumed to be those computed by the existing integrated severe accident analysis code—as well as its inference speed. Additionally, situational resilience may be incorporated to ensure that the surrogate model can operate smoothly on portable devices in scenarios where power and communications are completely disrupted.

2.3. Limitations of Surrogate Modelling based on Pure Data-driven Approach

Purely data-driven models—particularly deep learning (DL)—offer compelling predictive power but face notable shortcomings in safety-critical applications like severe accident analysis. Foremost is their lack of explainability: deep neural networks often function as “black boxes,” obscuring how millions of parameters combine to produce a given output. This interpretability gap complicates trust and acceptance, especially in regulated contexts where clear justification of model behavior is essential [4].

Equally important is the issue of data quality and quantity. Generating representative datasets for severe accidents is both expensive and time-consuming, making it challenging to capture the full spectrum of possible scenarios. When a model encounters conditions outside its training distribution—so-called extrapolation—it is difficult to predict how accurately it will perform. Consequently, purely data-driven models risk delivering unreliable results under off-design or extreme conditions, which is unacceptable for high-stakes decision-making. Embedding physical constraints or leveraging physics-based models can mitigate some of these issues by providing inductive biases and reducing reliance on prohibitively large, high-fidelity datasets.

3. Physics-Informed Neural Networks and its Application

3.1. Basic Concept of Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) are a methodological framework in which deep learning models are trained to directly satisfy physical laws (e.g., partial differential equations or conservation laws) [5]. Unlike conventional machine learning approaches that primarily rely on large-scale labeled data, PINNs incorporate the structural constraints of governing equations and initial or boundary conditions into the loss function. Specifically, the model’s predictions are evaluated against the underlying physical equations, and a term quantifying the discrepancy is included in the loss. During backpropagation, this term steers the optimization of network parameters in a physically consistent manner. Consequently, even in scenarios where observational data are scarce or difficult to obtain, PINNs—potentially integrated with numerical methods—facilitate the discovery of solutions that remain consistent with physical principles, thereby enabling more accurate and efficient estimation of complex physical systems.

3.2. Transient Heat Flow in a Semi-Infinite Heat Slab

Transient heat flow in a semi-infinite heat slab constitutes MELCOR's Gedanken C problem. To evaluate the MELCOR heat conduction models, predictions for heat conduction in a solid are benchmarked against the exact analytical solution for transient heat transfer in a semi-infinite solid with convective boundary conditions [7].

In MELCOR, transient heat flow in a semi-infinite solid with convective boundary conditions is approximated using a 10 m-thick finite slab. The slab’s nodes are distributed logarithmically, with the smallest spacing on the left side, which faces a large control volume and has a convective boundary condition (heat transfer coefficient of 10 W/m²K). An adiabatic boundary condition is applied on the slab's right side, effectively simulating a semi-infinite domain.

3.3. PINN Formulation for the Semi-Infinite Heat Slab Problem

Building on the transient heat flow problem described above, we now illustrate how a Physics-Informed Neural Network (PINN) can be formulated and trained to approximate the solution for heat conduction in a semi-infinite slab. The semi-infinite domain is approximated in MELCOR by a 10 m-thick finite slab with logarithmic nodal spacing; here, we use PINNs to directly encode the governing heat conduction equation and boundary conditions into the network's loss function.

3.3.1. Governing Equations and Boundary Conditions

The transient, one-dimensional heat conduction in a semi-infinite slab (x ≥ 0) is governed by the following partial differential equation (PDE):

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

where $(T(x,t))$ is the temperature, (x) is the spatial coordinate, (t) is time, and (α) is the thermal diffusivity of the slab material. In MELCOR's semi-infinite slab problem, the left boundary $((x = 0))$ is subject to a convective boundary condition:

$$-k \left.\frac{\partial T}{\partial x}\right|_{x=0} = h\left(T(0,t) - T_\infty\right)$$

where (k) is the thermal conductivity of the slab, \(h\) is the heat transfer coefficient (10 W/m²K in this example), and (T_∞) is the ambient temperature of the control volume on the left side. Meanwhile, the right boundary $((x = 10\ \mathrm{m}))$ is assumed to be adiabatic:

$$[\left.\frac{\partial T}{\partial x}\right|_{x=10} = 0.]$$

An initial condition $(T(x,0) = T_0)$ for the entire slab completes the problem specification. These conditions collectively reproduce the physical setup of a semi-infinite domain for short- to medium-duration transients, as the temperature gradients at the right boundary remain minimal if sufficiently insulated and the analysis timeframe is not too long.

3.3.2. Loss Function Construction in PINNs

A key distinction of PINNs is that they embed the PDE and boundary conditions directly into the loss function. Specifically, the total loss $(\mathcal{L})$ typically comprises three main components:

1. Data Loss $(\mathcal{L}_{dt})$:
 If any temperature data points $((x_i, t_i, T_i))$ are available (e.g., from sensor measurements or high-fidelity simulations), the model predictions $(\hat{T}(x_i, t_i))$ are penalized via a mean squared error (MSE) term:

$$\mathcal{L}_{dt} = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left(\hat{T}(x_i, t_i) - T_i\right)^2.$$

2. PDE $Loss(\mathcal{L}\square\mathcal{PE})$:
 The residual of the PDE is enforced by sampling collocation points (x_j, t_j) in the interior of the domain and time span. The partial derivatives $\partial\hat{T}/\partial t$ and $(\partial^2\hat{T}/\partial x^2)$ are computed via automatic differentiation, and the MSE of the PDE residual is added to the loss:

$$\mathcal{L}\square\mathcal{PE} = \frac{1}{N_{\text{PDE}}} \sum_{j=1}^{N_{\text{PDE}}} \left(\frac{\partial \hat{T}}{\partial t}(x_j, t_j) - \alpha \frac{\partial^2 \hat{T}}{\partial x^2}(x_j, t_j)\right)^2$$

3. Boundary and Initial Condition Loss $(\mathcal{L}_{\mathcal{B},\mathcal{I}})$:
 For boundary conditions, points are placed at $(x = 0)$ and $(x = 10\ \mathrm{m})$. The PINN is trained to satisfy:

$$[-k \left.\frac{\partial \hat{T}}{\partial x}\right|_{x=0} - h\left(\hat{T}(0,t) - T_\infty\right) = 0, \quad \left.\frac{\partial \hat{T}}{\partial x}\right|_{x=10} = 0.]$$

 Similarly, for the initial condition, $(T(x,0) = T_0)$. The MSEs of these constraints at designated collocation points along the boundaries and initial plane are added to the total loss.

Hence, the overall loss function is:

$$\mathcal{L} = \omega_{\text{data}}\, \mathcal{L}_{dt} + \omega_{\text{PDE}}\, \mathcal{L}\square\mathcal{PE} + \omega_{\text{BC,IC}}\, \mathcal{L}_{\mathcal{B},\mathcal{I}},$$

where (ω_{data}), (ω_{PDE}), and $(\omega_{\text{BC,IC}})$ are weighting factors that balance the influence of data fitting versus physical constraints.

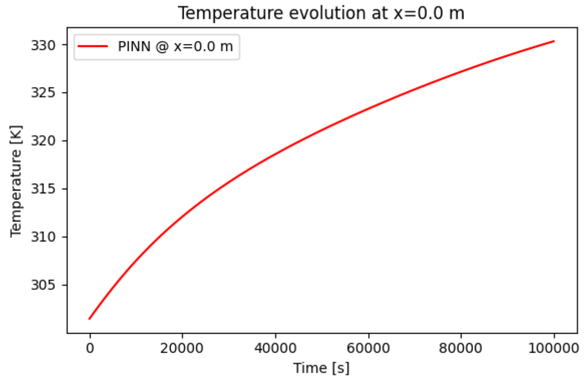
3.3.3. Training and Evaluation

Once the loss function is specified, the network can be trained via gradient-based optimizers (e.g., AdamW, L-BFGS) with automatic differentiation enabling exact gradients of $(\hat{T})$ with respect to both space and time. The trained PINN yields a continuous mapping $((x,t) \mapsto T)$, which can be evaluated at any point in the domain without the need for explicit spatial discretization.

To gauge its performance, the PINN solution can be compared against:

1. Analytical Solutions: For a semi-infinite slab with a convective boundary at $(x = 0)$, well-known solutions (e.g., those involving the error function $(\mathrm{erf}(\cdot))$ for simpler boundary conditions) or published references can be used to assess accuracy.
2. Numerical Solutions: MELCOR's discretized model provides nodal temperature values over time. Comparing the PINN-predicted temperature profiles with MELCOR

solutions, especially near the left boundary where large temperature gradients occur, offers a direct validation of how well the PINN replicates high-fidelity severe accident analysis code outputs.



3.3.4. Practical Considerations in PINN Deployment

Although PINNs naturally incorporate physical constraints, a few practical considerations arise in using them for real-world problems:

- Hyperparameter Selection: Choosing appropriate network depth, width, activation functions, and weighting coefficients (ω) can significantly impact convergence and stability.
- Computational Cost: While PINNs can offer smooth, mesh-free solutions, training can be computationally intensive if the domain or timescale is large. Efficient sampling strategies and adaptive methods (e.g., adaptive collocation point selection) may be necessary to scale up to more complex problems in severe accident scenarios.

4. Discussion of Technical and Practical Constraints

In this section, we broaden the perspective beyond the heat slab example to address the overarching limitations and challenges associated with applying deep learning—and particularly PINNs—to severe accident analysis. These issues encompass scaling to high-dimensional systems, ensuring numerical stability and interpretability, and meeting the real-time or near real-time computational requirements often demanded by safety-critical applications. Furthermore, questions of regulatory acceptance and validation remain open, underscoring the importance of transparent model development and robust uncertainty quantification.

By exploring both the strengths and weaknesses of purely data-driven DNNs and physics-informed approaches, we aim to map out a balanced research roadmap—one that leverages the complementary capabilities of traditional numerical solvers and advanced machine-learning frameworks to achieve both accuracy and computational feasibility in the context of severe accident management.

REFERENCES

[1] *Status and Evaluation of Severe Accident Simulation Codes for Water Cooled Reactors*. Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2019.

[2] Y. Lee *et al.*, "Surrogate model for predicting severe accident progression in nuclear power plant using deep learning methods and Rolling-Window forecast," *Annals of Nuclear Energy*, vol. 208, p. 110816, 2024/12/01/ 2024, doi: https://doi.org/10.1016/j.anucene.2024.110816 .

[3] S. H. Song *et al.*, "Application of reinforcement learning to deduce nuclear power plant severe accident scenario," *Annals of Nuclear Energy*, vol. 205, p. 110605, 2024/09/15/ 2024, doi: https://doi.org/10.1016/j.anucene.2024.110605 .

[4] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue*, vol. 16, no. 3, pp. 31-57, 2018.

[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686-707, 2019.

[6] M. J. Asher, B. F. W. Croke, A. J. Jakeman, and L. J. M. Peeters, "A review of surrogate models and their application to groundwater modeling," *Water Resources Research*, vol. 51, no. 8, pp. 5957-5973, 2015, doi: https://doi.org/10.1002/2015WR016967.

[7] L. Humphries *et al.*, "MELCOR Computer Code Manuals-Vol. 3: MELCOR Assessment Problems, Version 2.1 6850 2015," *SAND2015-6693 R, Sandia National Laboratories, Albuquerque, NM*, 2015.