Research on Surrogate Model for Thermal-Hydraulic Codes using LightGBM

Jiwon Han [a], Hyojun Yi [a,b], Hyeonmin Kim [c], Seunghyoung Ryu [a,b*]
[a]*Sejong University*
[b]*Artificial Intelligence and Robotics Institute*
[c]*Korea Atomic Energy Research Institute*
[*]*Corresponding author: shryu@sejong.ac.kr*

Keywords : machine learning, surrogate model, LightGBM, thermal-hydraulic code, probabilistic safety assessment

1. Introduction

Probabilistic safety assessment (PSA) is a methodology for quantifying the risk of a nuclear power plant (NPP) by estimating the likelihood and consequences of accident scenarios. In this process, thermal-hydraulic (TH) codes play a key role by simulating the NPP's TH behavior throughout accident progression. However, TH codes require solving nonlinear partial differential equations, making even a single scenario computationally expensive. To overcome this bottleneck and evaluate large-scale scenarios, surrogate models are needed. Existing research on data-driven surrogate models has largely relied on deep learning [1] due to its ability to model highly nonlinear relationships. Interestingly, recent studies have shown that tree-based models often outperform deep learning methods on typical tabular datasets composed of structured variables [2]. This observation is relevant to TH surrogate modeling, where accident scenario information is often organized as structured, tabular inputs. Despite their suitability for tabular inputs, tree-based methods have not been widely adopted in surrogate modeling research because they are primarily designed for univariate targets; therefore, they are difficult to apply to complex, multivariate sequence prediction required for TH codes.

In light of this, we propose a method for applying tree-based machine learning models to TH code sequence prediction. Specifically, we use LightGBM (LGBM) [3] and compare the prediction performance of LGBM- and random forest (RF)-based surrogate models [4] with deep learning baselines.

2. Methods

The accident scenario vector $\mathbf{x}$ consists of 27 scenario parameters, including, for example, the opening degree of valve and the timing of operation. The target $Y(t)$ includes the pressurizer pressure (PPS) and peak cladding temperature (TCRHOT), which are time-series outputs of the TH code. Since tree-based regressors are typically single-output, we train separate models for PPS and TCRHOT, and we reconstruct trajectories by evaluating the models with a time-conditioned input. We learn the mapping $f: \mathbf{x} \rightarrow Y(t)$ and compare Tree-based Models and Deep Learning-based Models.

2.1 Tree-based Models

A decision tree is a rule-based model that learns a set of hierarchical decision rules to partition the input space and make prediction. This structure allows decision trees to capture nonlinear relationships between input and output variables. RF is an ensemble method that builds multiple decision trees using bootstrap-resampled data and random subsets of features [4]. Predictions are obtained by aggregating the outputs of individual trees (e.g., averaging for regression). This ensemble strategy typically improves robustness and reduces variance compared to a single decision tree.

Gradient boosted decision tree (GBDT) is a boosting-based ensemble method in which trees are added sequentially to fit the negative gradients of the loss function with respect to the current model. LGBM is a high-performance GBDT implementation designed for efficient training and inference [3], and it has demonstrated strong performance across a wide range of tabular learning tasks.

2.2 Deep Learning-based Models

Feedforward Neural Network (FNN) is a fundamental neural model that represents a nonlinear mapping from inputs to outputs through successive transformations. In this architecture, information flows in a single direction-from input to output-without any feedback or recurrent connections. By composing multiple nonlinear transformations, an FNN can approximate complex input-output relationships and is commonly used as a general-purpose baseline for regression and prediction.

Recurrent Neural Network (RNN) is designed to process sequential data through recurrent connections. This structure retains information over time by maintaining internal hidden states that capture temporal dependencies from previous inputs. However, a conventional RNN often struggles to learn long-term dependencies due to the vanishing gradient problem. To overcome this limitation, Long Short-Term Memory (LSTM) networks were introduced in [5]. LSTM features a specialized memory cell structure controlled by gating mechanisms such as forget, input, and output gates, which selectively regulate the flow of information. This allows the model to effectively retain important long-term patterns, making it a standard baseline model for various time series applications.

2.3 Proposed Model

Fig. 1 depicts the overall architecture of the proposed LGBM-based surrogate model. The input to the LGBM consists of two parts: scenario descriptor $\mathbf{x} \in \mathbb{R}^{27}$ and a polynomial time encoding $\phi(t) \in \mathbb{R}^{3}$. These are concatenated to form a time-conditioned input vector $z(t) = [\mathbf{x}, \phi(t)] \in \mathbb{R}^{30}$. To accommodate the tabular input requirement of the tree-based model, the multi-dimensional dataset across all scenarios and time steps is flattened into a two-dimensional format. This flattened dataset is then used as input for LGBM to produce a pointwise prediction $\hat{Y}(t)$. By evaluating the model at all discrete time points, we could obtain the full output trajectories.

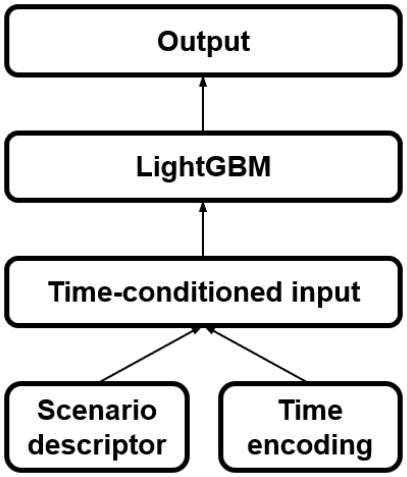

Fig. 1. Overall architecture of the proposed model.

3. Experiments

3.1 Experimental Setup

The dataset consists of 10,000 input-output pairs simulated using the MAAP TH code. To ensure a fair comparison between decision tree-based and neural network-based models, normalization was applied to both inputs and outputs. The models were trained and evaluated using 5-fold cross-validation. Grid search was applied for hyperparameter selection for all models. The model performance was evaluated using the root mean square error (RMSE) and mean absolute error (MAE).

3.2 Predictive Performance

We compared the performance of the proposed LGBM model with FNN, LSTM, and RF-based models. Table I summarizes the RMSE and MAE (mean±std) for PPS and TCRHOT. As shown in the table, LGBM achieved superior predictive performance across all targets. Specifically, LGBM achieved an overall reduction of approximately 14.17% in average RMSE and 13.27% in average MAE across both target variables compared to the LSTM.

Table I: Comparison of RMSE and MAE (mean±std)

Model	PPS (RMSE)	TCRHOT (RMSE)
FNN	0.261±0.301	0.628±0.476
LSTM	0.240±0.310	0.635±0.506
RF	0.250±0.226	0.641±0.448
LGBM	**0.174**±0.175	**0.577**±0.442
Model	**PPS (MAE)**	**TCRHOT (MAE)**
FNN	0.174±0.218	0.467±0.383
LSTM	0.152±0.220	0.451±0.398
RF	0.153±0.157	0.475±0.361
LGBM	**0.101**±0.126	**0.422**±0.354

4. Conclusions

In this study, we compare the accuracy of a TH code surrogate model with a conventional deep learning-based model and a tree-based model. To predict TH trajectories, we apply time-conditioned input formulation to the LGBM. Experimental results show that the proposed LGBM model achieves superior prediction performance, reducing the average RMSE by 14.17% and the average MAE by 13.27% compared to the LSTM model.

ACKNOWLEDGMENT

This work was partially supported in part by the National Research Council of Science & Technology (NST) grant from the Korea government (MSIT) (No. GTL24031-000), in part by the IITP (Institute of Information & Communications Technology Planning & Evaluation) - ICAN (ICT Challenge and Advanced Network of HRD) grant funded by the Korea government (Ministry of Science and ICT) (IITP-2026-RS-2024-00436528), and in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00454514)

REFERENCES

[1] Ryu, Seunghyoung, et al. "Probabilistic deep learning model as a tool for supporting the fast simulation of a thermal–hydraulic code." Expert Systems with Applications 200 (2022): 116966.
[2] Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux. "Why do tree-based models still outperform deep learning on typical tabular data?." Advances in neural information processing systems 35 (2022): 507-520.
[3] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).
[4] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[5] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.