Hybrid Deep Learning-Based Surrogate Model for Thermal-Hydraulic Codes with LSTM, CNN, and Transformer

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

***Keywords :** deep learning, surrogate model, accident scenarios, probabilistic safety assessment, thermal-hydraulic code

1. Introduction

An accident scenario refers to a sequence of various events occurring during the operation of a nuclear power plant. Given a specific accident scenario, it is possible to simulate changes in system states (e.g., temperature, pressure) using a thermal-hydraulic (TH) code. Based on the simulation, probabilistic safety assessments (PSA) evaluate the safety of the nuclear power plant by estimating the outcome (reactor core damage) of the accident scenario. The analysis of safety can be enhanced through simulations of a larger number of scenarios. However, the computing time, which ranges from minutes to hours, limits the feasibility of this massive analysis [1]. Deep learning models require a long training time but have a low computational cost during inference. This makes it possible to use deep learning models as surrogate models for TH code. The previous work of [2] proposed an LSTM-based surrogate model and demonstrated the potential of deep learning-based surrogate models. However, there is still room for improvement in terms of accuracy. To address this, we propose a hybrid deep-learning model that integrates LSTM, CNN, and Transformer.

2. Problem Formulation

Following the problem formulation described in [2], each accident scenario consists of the operator's response and system control information (e.g., the degree of opening of a specific valve, the operation time of a switch). This information can be represented as a d-dimensional vector $\mathbf{x} = [x_1, x_2, \cdots, x_d]^T \in \mathbb{R}^d$. The deep learning surrogate model takes accident scenarios as input and predicts values of key variables, such as core temperature and pipe pressure, which are the results of the TH code. Therefore, for a prediction window T, the output is time series data of d' variables, which can be defined as $\mathbf{Y} = [y_1, y_2, \cdots, y_T]^T \in \mathbb{R}^{T\times d'}$.

3. Deep Learning Model

The proposed LCT model is a structured combination of LSTM, Transformer, and CNN, leveraging the strengths of each. Below subsection describes each layer and LCT in detail.

3.1 LSTM

A recurrent neural network (RNN) is a class of neural network architecture that receives output or hidden states of the previous time-step as an input of the current time-step [2]. The standard RNN structure suffers from the vanishing gradient problem, which led to the introduction of Long Short-Term Memory (LSTM). LSTM uses a gating mechanism to retain important information and discard unnecessary information over time. By leveraging input, forget and output gates, LSTM controls the flow of information along the time axis and effectively learns long-term dependencies.

3.2 1D-Convolution

The convolutional layer is a fundamental component of convolutional networks, typically used for processing images. A 1D convolution (Conv1D) is a variation of the convolutional layer designed for sequential data. Conv1D applies a set of learnable filters across the input sequence along the time dimension. Therefore, it is possible to capture short-term temporal features.

3.3 Transformer

Transformer (TR) utilizes the attention mechanism, which directly learns the relationships across all time steps, effectively addressing the long-term dependency problem of RNN. Given an input sequence, the self-attention mechanism computes three metrices: Query (Q), Key (K) and Value (V), all of which are derived from the input using learned weight matrices. Then, attention scores are calculated using scaled dot-product in Equation (1) [3].

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

3.4 Weighted Positional Encoding

Positional encoding (PE) was introduced in transformer models to inject order information into sequence data [3]. In general, a sinusoidal function is used to generate the PE vector. This PE vector is concatenated with the scenario vector **x,** which becomes a sequential input to the surrogate model [2]. To enhance the time-dependent capabilities, we propose a weighted positional encoding (WPE) by multiplying position-wise weights with the PE vector, as described in Equation (2).

$$\begin{cases} PE_{(pos,2i)} = w_{pos} \cdot \sin\left(\frac{pos}{10000^{2i/d}}\right) \\ PE_{(pos,2i+1)} = w_{pos} \cdot \cos\left(\frac{pos}{10000^{2i/d}}\right) \end{cases} \quad (2)$$

3.5 Model Structure

The overall model structure is as shown in Figure 1. The accident scenario vector $\boldsymbol{x}$ is converted into the latent vector $\boldsymbol{z}_t$ through WPE and LSTM. Then, it passes through the ConvTr block consisting of Conv1D, TR, and deconvolution (deConv) layers. By combining Conv1D and Tr layers, the ConvTr block effectively captures both local and global dependencies. In order to apply residual connection, deConv layer is used to match the output shape of the ConvTr block [4]. The final output is obtained through the LSTM and Linear layers.

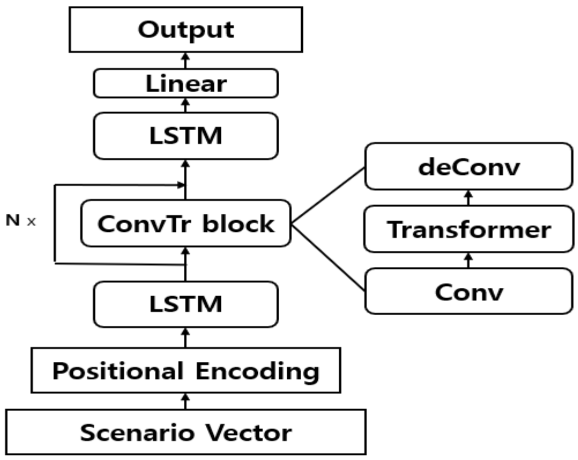

Fig. 1. LCT Model Structure

4. Experiments

The proposed model is trained and evaluated with MAAP code data on loss of offsite power (LOOP) and station black out (SBO) cases. The number of data is 10,000, and hyperparameter are set through grid search using a randomly sampled training (n=3,000) and validation set (n=2,000), with the selected values described in Table I.

Table I: Hyperparameter setting

HyperParameter			**Number of Layers**	
LR	BS	Epoch	LSTM	ConvTr
1e-3	16	30	2	2
Loss Function			**Optimizer**	
Mean Squared Error (MSE)			Adam AdamW(for LCT)	

To evaluate the performance of the proposed model, we compared the prediction accuracy of the PPZ variable with three baseline models (FNN, RNN, and TR). The experimental results are validated through 5-fold cross-validation for all 10,000 scenarios. Table II presents the averages of the statistics across all folds and random seed settings. As can be seen in the table, the LCT model shows the lowest error statistics, with MAE and MAPE decreased by 7% and 34% compared to the TR.

Table II: Comparison of prediction errors on PPZ

Model	**MAE**			**MAPE(%)**		
	Mean	Med	Stdev	Mean	Med	Stdev
FNN	0.261	0.155	<u>0.301</u>	105.8	42.1	241.3
RNN	0.243	<u>0.102</u>	0.332	96.5	36.7	230.2
TR	<u>0.243</u>	0.109	**0.297**	<u>90.9</u>	<u>35.9</u>	<u>193.2</u>
LCT	**0.226**	**0.091**	0.320	**60.1**	**28.8**	**179.9**

5. Conclusions

In this research, we proposed the LCT model, a hybrid of LSTM, CNN, and Transformer for the surrogate model of TH code. To improve prediction accuracy, we incorporated Weighted Positional Encoding (WPE), which allows the model to learn the relative order of the input sequence. Additionally, the CNN and Transformer-based ConvTr block efficiently captures both short-term and long-term features. The model's performance was evaluated through experiments, showing a significant improvement in prediction accuracy. Compared to transformer model, our proposed model reduced MAE by approximately 7% and MAPE by 34%.

ACKNOWLEDGMENT

This research was partially supported by the National Research Council of Science & Technology (NST) grant from the Korea government (MSIT) (No. GTL24031-000) and 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-2025-RS-2024-00436528).

REFERENCES

[1] Kim, Hyeonmin, Jaehyun Cho, and Jinkyun Park. "Application of a deep learning technique to the development of a fast accident scenario identifier." *IEEE Access* 8 (2020): 177363-177373.
[2] 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.
[3] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
[4] He, Shujun, et al. "RNAdegformer: accurate prediction of mRNA degradation at nucleotide resolution with deep learning." *Briefings in Bioinformatics* 24.1 (2023): bbac581.
[5] Zhao, Ziqi, et al. "Interpretation of time-series deep models: A survey." *arXiv preprint arXiv:2305.14582* (2023).