

Article

A Transformer-VAE Framework with Knowledge Distillation for Fast Prediction of Nuclear Power Plant Accident Transient Response

Bo Pang , Yuanfeng Lin, Guoxu Qin , Siyuan Zhang, Zhixin Pang, Yaoyi Zhang, Di Chen, Guohai Cao and Qingzhong Ai

National Key Laboratory of Nuclear Reactor Technology, Nuclear Power Institute of China, Chengdu 610213, China; cannonlin@hotmail.com (Y.L.); qgx1874@163.com (G.Q.); siyuan_tooru@163.com (S.Z.); pangzhixin2020@163.com (Z.P.); zhangyymath@163.com (Y.Z.); chendi1995425@gmail.com (D.C.); cwgh2014@163.com (G.C.); aqz1995@163.com (Q.A.)

* Correspondence: pb2180757@163.com

Abstract

Conventional analysis of nuclear power plant accident transient responses heavily relies on physical simulation programs, whose computational time significantly exceeds the actual accident response duration, thereby severely hindering real-time safety assessment. This paper proposes a novel framework that integrates a Transformer-based Variational Autoencoder (VAE) with knowledge distillation for the fast prediction of nuclear power plant (NPP) accident transient responses. The approach involves constructing a latent space to extract essential features from transient response data using an Encoder–Decoder model based on the VAE architecture. A key innovation is the establishment of a direct mapping between plant operating condition parameters and the latent space, enabling the one-step generation of accident transients without iterative sequential prediction. The Encoder and Decoder leverage Self-Attention and Cross-Attention mechanisms to enhance feature extraction and conditional generation. Furthermore, the Encoder is distilled into a Mapper network, which predicts the latent features directly from the operating conditions, resulting in an efficient Mapper–Decoder pipeline for rapid prediction. The proposed model was evaluated against traditional Long Short-Term Memory (LSTM) and fully connected neural networks. Experimental results demonstrate that the proposed framework achieves superior performance in predicting NPP accident transients, indicating its strong potential for efficient safety analysis and system design optimization.

Keywords: transformer; VAE; attention; nuclear power plant simulator



Academic Editor: Li Xi

Received: 17 June 2026

Revised: 5 July 2026

Accepted: 7 July 2026

Published: 13 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The global nuclear energy sector is experiencing significant growth, driven in part by the integration of advanced technologies like artificial intelligence (AI). AI is increasingly being applied in various sub-fields such as the design and operational management of nuclear power plants. The deep integration of AI and nuclear energy is poised to become a major trend in the future development of nuclear power.

The recognition and prediction of accidents in nuclear power plants (NPPs) have been significantly advanced by applying various deep learning techniques. For instance, deep-learning-based accident recognition systems assist operators in identifying anomalies [1].

Long Short-Term Memory (LSTM) networks are employed to predict sensor readings during normal operation; deviations from actual readings beyond a set threshold help detect abnormal events [2]. Further, a Cost-Sensitive LSTM (CLSTM) has been developed to identify anomalous states in shipboard pressurized water reactors [3], while a CNN–LSTM hybrid model, trained on VVER-1000 simulation data, achieves efficient and accurate predictions of plant transients [4]. Beyond accident recognition, neural networks facilitate the construction of virtual sensors using operational data for equipment aging assessment [5]. Unsupervised approaches, such as those combining LSTM with Variational Autoencoder (VAE), have been applied to evaluate unknown events [6], and hybrid accident simulations are enhanced through neural network integration [7]. Additionally, neural networks have been utilized for core optimization tasks, such as rapidly estimating the local power peaking factor (LPPF) in boiling water reactors (BWRs) with minimal error compared to reference codes like HELIOS [8]. Physics-Informed Neural Networks (PINNs) incorporate physical laws into accident simulations [9], and nonlinear autoregressive neural networks are used to predict pipe fracture sizes in loss-of-coolant accidents (LOCA) [10]. Unsupervised frameworks using VAE and Isolation Forest enable abnormal state detection by training exclusively on normal operational data [11,12]. To sum up, the application of deep learning in nuclear power plant accident prediction and safety analysis has demonstrated significant potential to shift the paradigm from passive response to proactive early warning, laying a solid technical foundation for further exploration of intelligent operation and real-time decision support [4].

This study aims to overcome the computational bottlenecks of physical-model-based accident simulation in nuclear power plants, which are prohibitive for real-time application and training. The methodology involves two key steps: first, generating a comprehensive set of transient response data through conventional physical simulation programs; second, constructing a high-fidelity time-series model based on plant operating conditions using Transformer and VAE neural networks. The resulting model achieves prediction accuracy comparable to that of physical simulations, with a significantly reduced computational cost.

2. Methodology

2.1. Data Collection

In this work, we use the RELAP5 [13] simulation code to obtain a large number of simulation programs for nuclear power plant accidents, consisting of 42 conditional parameters that determine the transient response to a nuclear power plant accident. As shown in Table 1, core cladding peak temperature, cladding surface temperature, pressurizer water level, and primary circuit pressure, these four parameters constitute the time-series output data, and the remaining 38 parameters form the conditional parameters. The Encoder takes the 38 conditional parameters as input, while the only conditional input to the Decoder is time. This design allows the latent space vector to encode complete transient information, with time only used to define the time span of the reconstructed time-series data. The detailed definitions and units of these four output parameters are summarized in Table 2.

2.2. Data Preprocessing

2.2.1. Linear Normalization

Normalization of the dataset is required for the speed of convergence of the model:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X_{norm} is the normalized data, X , X_{max} , X_{min} are each data point, the maximum value as well as the minimum value, respectively.

Table 1. Working condition parameters of the simulation code.

Time Series Data	The Conditional Parameters of the Encoder			The Condition Parameters of the Decoder
	time	Core operating power	decay heat	
Peak cladding temperature	Break size	Pressure in containment compartments at the breach location	Break discharge coefficient	
	Break flow friction coefficient	Initial pressure of the safety injection tank	Initial temperature of the safety injection tank	
Cladding surface temperature	Volume of safety injection tank	Water storage capacity of the safety injection tank	Trigger pressure of the safety injection tank	
	Area of injection line for safety injection tank	Injection friction coefficient of the safety injection tank	Local loss coefficient of safety injection tank	
	IRWST Initial pressure	Initial temperature of IRWST [14]	Tank volume of IRWST	time
Pressurizer water level	Trigger pressure of IRWST	Area of injected pipeline of IRWST	Injection friction coefficient of IRWST	
	Local friction factor of IRWST	Discharge coefficient of ADS-1	Discharge coefficient of ADS-2	
	Discharge coefficient of ADS-3	Discharge coefficient of ADS-4	Open time of ADS-1	
Primary circuit pressure	Open time of ADS-2	Open time of ADS-3	Open time of ADS-4	
	Friction coefficient of ADS-1~ADS-3	Initial pressure of CMT	Initial temperature of CMT	
	Volume of CMT	Trigger pressure of CMT	Area of injected pipeline of CMT	
	Injection friction coefficient of CMT	Local friction coefficient of CMT		

Table 2. Transient safety parameters.

Parameters	Definition	Unit
T_c	Peak cladding temperature	°C
T_s	Cladding surface temperature	°C
RL	The water level of the pressurizer	m
P	The pressure of the primary circuit	MPa

2.2.2. Training Dataset

In this paper, the three models of Encoder, Decoder, and Mapper are constructed for one-step predictive generation. The dataset does not need to be sliced into subsequences and only requires shuffling and normalization. Each 500-s accident transient scenario

defined by the 38 conditional parameters constitutes one training sample. The inputs and outputs of the three models are summarized in Table 3.

Table 3. The inputs and outputs of the neural network.

	Mapper–Decoder	LSTM	Dense
Inputs	Condition parameters of the Encoder (in addition to times)	Condition parameters of the Encoder	Condition parameters of the Encoder
Outputs	Time series data	Time series data	Data of a single time node

As shown in the table, to validate the performance of the model, an LSTM and a fully connected network are added for comparison. LSTM is a variant of a recurrent neural network (RNN) [15]. All baseline models are trained using a strategy similar to that of the Encoder–Decoder framework, taking the Encoder’s conditional parameters as input to predict the accident transient response. The fully connected network [16] uses the Encoder condition parameter of a single time node as input data to predict the transient response at the corresponding time point. A part of the dataset is not used for training and is used to compare the generalization performance of the Mapper–Decoder [17], LSTM, and Fully Connected Networks.

2.3. Evaluation Metrics

We use MSE and R^2 as evaluation metrics for the model:

$$MSE(pred, true) = \frac{1}{n} \sum_{i=1}^n (pred_i - true_i)^2 \quad (2)$$

$$R^2(pred, true) = 1 - \frac{\sum_{i=1}^n (pred_i - true_i)^2}{\sum_{i=1}^n (\overline{true} - true_i)^2} \quad (3)$$

MSE directly quantifies the error between model predictions and the ground truth: the lower the MSE , the better the model’s fit to the primary circuit pressure. R^2 reflects the proportion of variance in the experimental data explained by the model; the closer R^2 is to 1, the more accurate the model’s predictions.

2.4. Kullback–Leibler (KL) Divergence

In the VAE structure, KL divergence is used to ensure the generative power of VAE:

$$KL(\mathcal{N}_1(\mu_1, \sigma_1^2), \mathcal{N}_2(\mu_2, \sigma_2^2)) = \log \frac{\sigma_2}{\sigma_1} + \frac{1}{2\sigma_2^2} (\sigma_1^2 + (\mu_1 - \mu_2)^2) - \frac{1}{2} \quad (4)$$

In VAE, $\mathcal{N}_2$ is defined as a standard normal distribution, $\mu_2 = 0$, $\sigma_2 = 1$. Meanwhile, the following changes are required to guarantee the gradient descent of the neural network.

$$KL(\mathcal{N}(\mu, \sigma^2), \mathcal{N}(0, 1)) = -\frac{1}{2} [1 + 2\log \sigma^2 - \mu^2 - \sigma^2] \quad (5)$$

The following software tools were used in this study: RELAP5/MOD3.3 for nuclear reactor transient simulation; Python 3.10 for data preprocessing; and PyTorch 2.1.1 for deep learning model implementation, including the Transformer-based Encoder–Decoder, Mapper network, and the LSTM and Dense baseline models.

3. Transformer-VAE

3.1. VAE

Autoencoder (AE) is commonly used in feature extraction engineering, which utilizes a “bottleneck” layer to filter the valid information [18]. It is able to extract valid features from larger data, mapping representational space data to latent space. VAE evolved on the basis of AE, and the Encoder of VAE no longer predicts the latent space vectors directly. Instead, it predicts the parameters of the distribution of the latent spatial vectors, where the latent space distribution functions are all normal functions [19]. The latent space vectors are obtained by the reparameterization trick, which is equivalent to the introduction of regularization for the latent space, making the Decoder’s decoding work with generative capabilities. The structures of AE and VAE are shown in Figure 1.

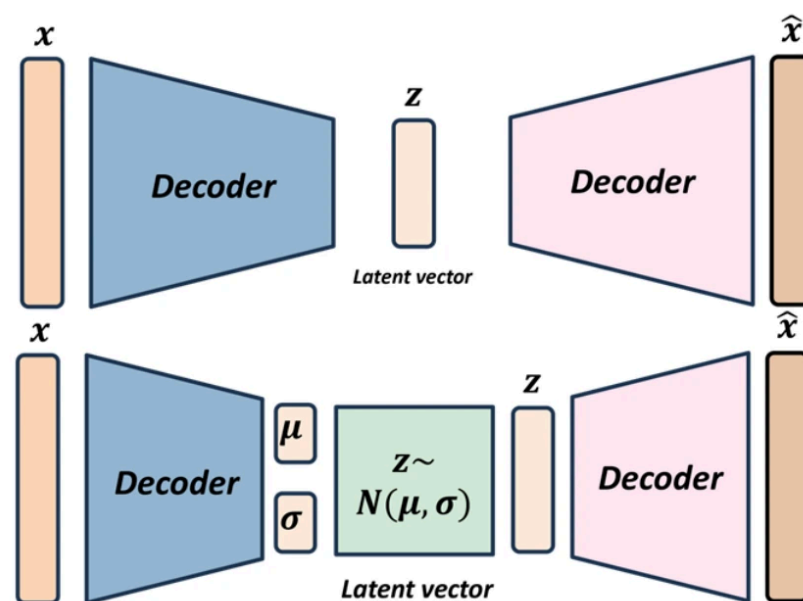


Figure 1. Autoencoder and Variational Autoencoder.

3.2. Network Building Block

VAE is used to create the latent space. As shown in Figure 2, the Attention layer and the feedforward network form a repeatable network block, which in turn builds the backbone network structure of the Encoder and Decoder. In order to prevent the problem of gradient vanishing due to the excess of network layers, the residual network structure [20] is used in it. The network block is the backbone structure that constitutes the Encoder–Decoder, but the Encoder and Decoder also have their own structures depending on the tasks.

Encoder–Decoder is connected in series, and the Encoder predicts the distribution function of vectors. After the reparameterization trick, the latent space vectors are reconstructed into the time series data by the Decoder. For the Encoder, as shown in Figure 3, its input conditions and the length of the feature vectors of the time series are not necessarily the length required by the Attention mechanism, so the data has to go through a layer of stretching first, which consists of the Dense layer. At the same time, the Attention mechanism cannot autonomously obtain the relative position of the time series data like time series neural networks (RNN, LSTM, GRU [21]). Therefore, the positional encoding is spliced into the input time series data to obtain the relative position information.

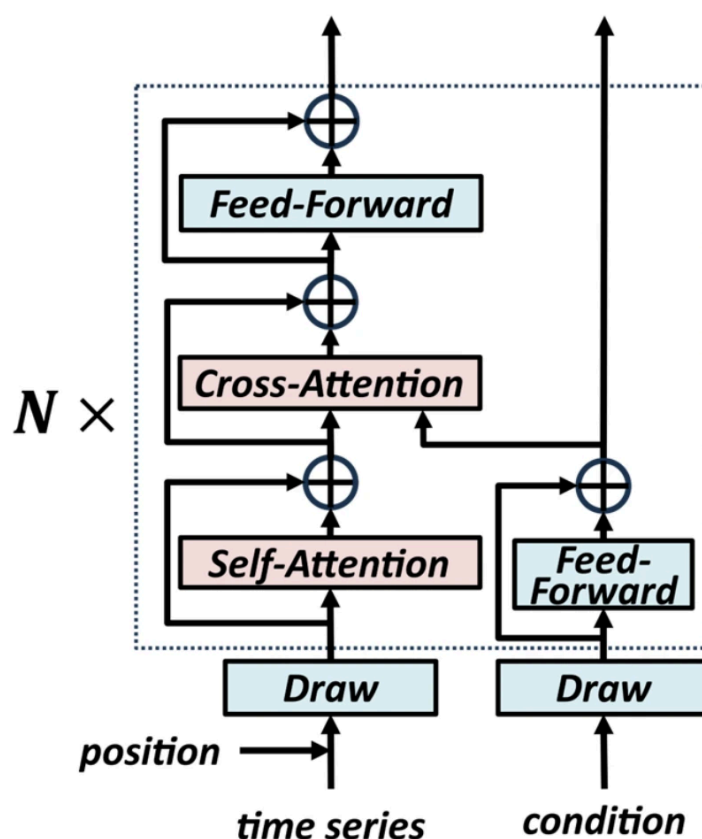


Figure 2. The main network block of the Encoder and Decoder.

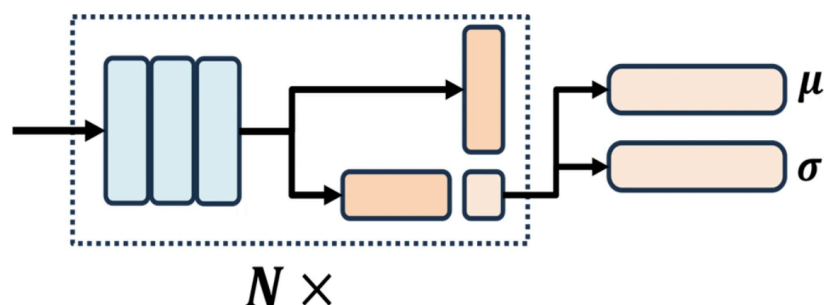


Figure 3. The network input structure of the Encoder.

A two-layer fully connected layer is added at the end of the network block in order to improve the fast-fitting ability of the backbone network. The number of cells in the first layer is regulated by dff , and nonlinearized by $ReLU$, and the second layer is a hyperparametric dimension of the key vector d_{model} control.

$$Y = Relu(XW_1 + B_1)W_2 + B_2 \quad (6)$$

The Encoder outputs two vectors via the multiple-head Attention structure, which represent the mean and variance of the latent Gaussian distribution, respectively, and its multiple-head structures output multiple values to form the mean and variance of the normal distribution. Its multiple-head structure outputs multiple values to form the mean and variance of a normal distribution, respectively. Decoder outputs time series data, which can be output directly. It is worth noting that the conditions of the Encoder and Decoder can be different.

3.3. Attention

To improve the generalization ability of the model, the Attention mechanism is used to enhance the model's ability to extract features. As shown in Figure 4, V , K , Q are the value vector, key vector, and query vector, respectively. Attention mechanism allows the model to apply Attention to more valuable data points and suppress useless noise. At the same time, there are some problems with the Attention mechanism, which has high computational complexity and memory usage.

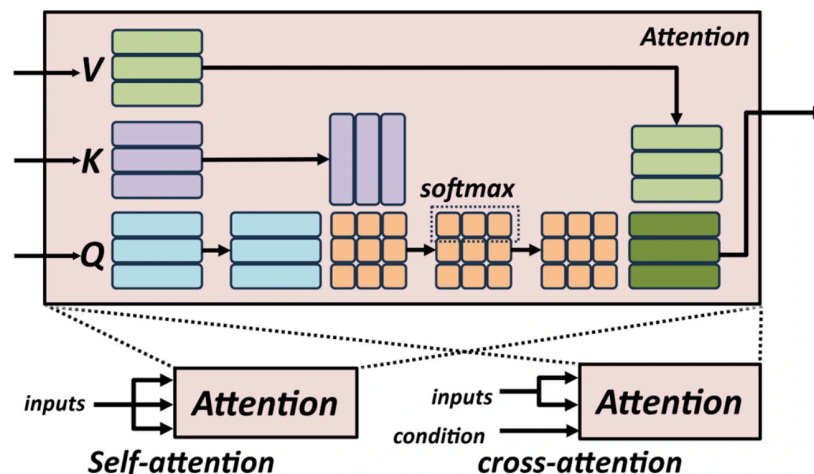


Figure 4. Self-Attention and Cross-Attention.

The formula for Attention is as follows:

$$Z = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) V \quad (7)$$

where Z is the output tensor, and d_k is the length of the dimension.

The Attention mechanism can be subdivided into Self-Attention and Cross-Attention. The difference between the two is that the input vectors are different; the input of Self-Attention consists of a set of vectors mapped into V , K , and Q through the Dense layer, respectively, while the Q of Cross-Attention is replaced by a conditional vector. Both Self-Attention and Cross-Attention are components of the network block. And Cross-Attention is used to introduce conditions.

$$V = w_1 i + b_1 \quad (8)$$

$$K = w_2 i + b_2 \quad (9)$$

$$Q = w_1 (i \text{ or } c) + b_1 \quad (10)$$

where i denotes the input tensor and c denotes the condition tensor.

3.4. Latent Vector and Position

The Attention architecture does not have the ability to obtain the relative location of individual temporal data. It is necessary to splice in the positional embedding at the input so that the model can obtain the relative location information. Figure 5 illustrates position embedding and splicing.

$$\vec{p}^{(i)} = \begin{cases} \sin(w_k \cdot t), & \text{if } i = 2k \\ \cos(w_k \cdot t), & \text{if } i = 2k + 1 \end{cases} \quad (11)$$

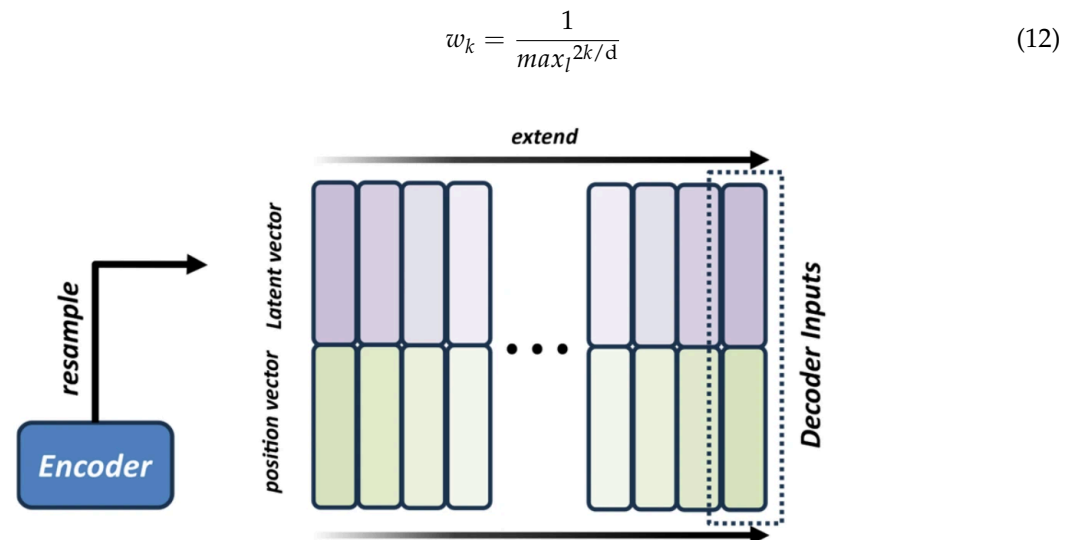


Figure 5. Position embedding and splicing.

As shown in Equation (11) The i value of the encoding vector located in the input feature vector can be obtained from Equation (1). $\max_l$ determines the maximum encoding length, and once the length of the input time series is exceeded, the positional encoding will be duplicated.

3.5. Knowledge Distillation

After the Encoder–Decoder training is complete, the Decoder is the model expected to reconstruct the time series data from the latent space. However, the trained Encoder cannot be directly used for inference because the complete time-series data of the accident transient is not available during the prediction phase. Therefore, knowledge distillation is performed on the trained Encoder to construct a mapper network, which only requires the conditional parameters as input (without the need for complete time-series data). In the Encoder–Mapper training, the Decoder is not involved; Encoder parameters are frozen, and only the Mapper’s network parameters are updated.

3.6. Loss Function

In Encoder–Decoder training, the loss function consists of two terms:

$$\text{loss} = \text{MSE}(X, X') + \beta \cdot \text{KL}\left(N(\mu, \sigma^2), N(0, 1)\right) \quad (13)$$

where MSE represents the reconstruction loss, which quantifies the reconstruction quality of the Decoder. KL denotes the Kullback–Leibler divergence, which regularizes the latent vectors generated by the Encoder to approximate the standard normal distribution. Removing the KL term will cause the model to degenerate from a Variational Autoencoder (VAE) to a vanilla Autoencoder (AE). Specifically, the KL divergence serves as a regularization constraint to ensure the smoothness and continuity of the latent space, which is essential for the generative capability of the VAE. β is a weighting hyperparameter that balances the two loss components: a larger β enforces the latent distribution to better align with the standard normal distribution.

In Encoder–Mapper training, the loss function is directly composed of the MSE:

$$\text{loss} = \text{MSE}(y_{\text{decoder}}, y_{\text{mapper}}) \quad (14)$$

where y_{decoder} and y_{mapper} are the output values of Encoder and Mapper, respectively. During training, the network parameters of the Encoder will not be updated, and the

output value of the Mapper will gradually approximate that of the Encoder. It is worth mentioning that loss compares the similarity between the two; there is no fixed labeled value, and the output of the Encoder contains noise. The absolute value of this loss function cannot fully reflect the convergence of the knowledge distillation process.

4. Training

4.1. Model Parameter Selection

As shown in Tables 4–6, we set the parameters for Encoder, Decoder, Mapper, LSTM, and Dense for the comparison.

Table 4. The network structure parameters of Encoder–Decoder–Mapper.

Encoder or Decoder		Mapper	
Max Length	1024	Time series data	Data of a single time node
dimension of the key vector	64	Activation	ReLU
Number of Attention heads	8	Dense-1	64
The unit of the Feedforward layer	32	Dense-2	64
Layers	1	Dense-3	64
Dropout rate	0.1	Add-1	Dense-1, Dense-3
β	0.002	Dense-4	64
		Dense-5	64
		Add-2	Add-1, Dense-5
		Dense-5	20

Table 5. The network structure parameters of LSTM.

LSTM	
Activation	Tanh
LSTM-1	128
LSTM-2	128
LSTM-3	128
Dense-4	128
Dense-5	4

Table 6. The network structure parameters of Dense.

Dense	
Activation	ReLU
Dense-1	128
Dense-2	128
Dense-3	128
Dense-4	128
Dense-5	4

4.2. Train Parameter

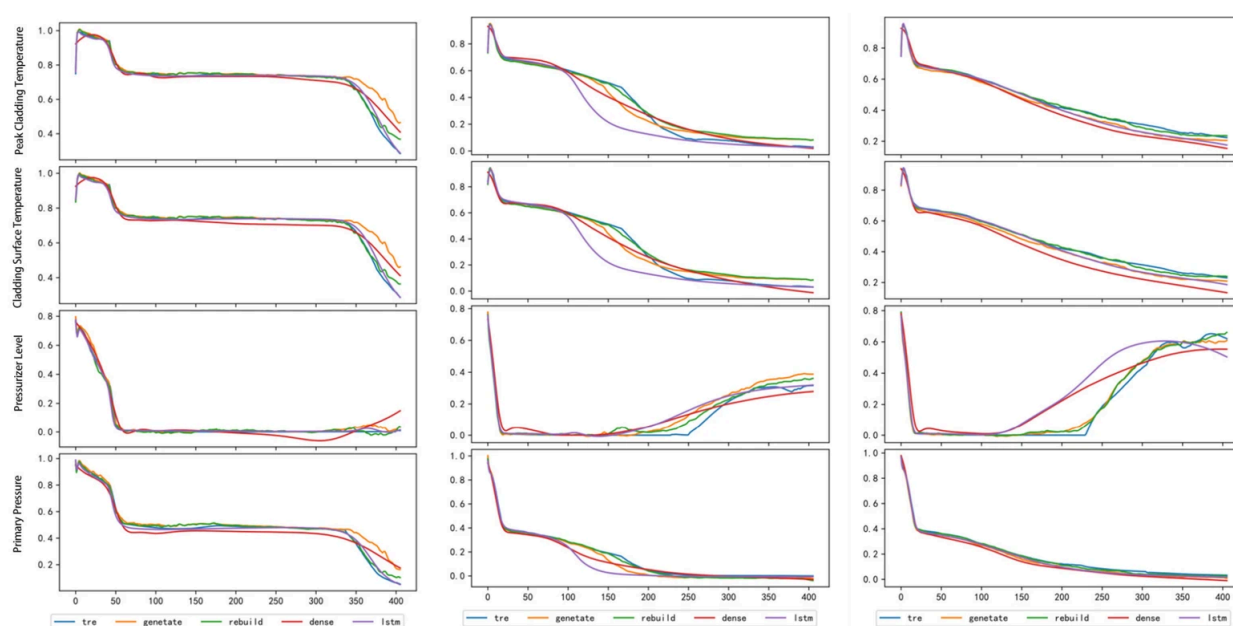
The stochastic sampling step introduces variance in the Encoder’s output at each training iteration, meaning the Mapper receives a noisy and non-deterministic input signal. To compensate for this uncertainty and learn a stable mapping, the Mapper–Decoder requires a longer training schedule (700 epochs) compared to models without stochastic components. In contrast, LSTM and Dense models converge within 100 epochs, as their forward passes are fully deterministic. The complete training parameters for the four models are summarized in Table 7.

Table 7. Training parameters of models.

	Encoder–Decoder	Mapper–Decoder	LSTM	Dense
Optimizer		Adam		
Epochs	100	700	100	100

5. Neural Network Model Prediction

Figure 6 shows the calculation results of some test sets. It shows the reconstruction results (rebuild) of Encoder–Decoder and the generation results (generate) of Mapper–Decoder for the four time series data of PCP , RL , T_c , and T_s , as well as the prediction results of LSTM and fully connected network.

**Figure 6.** Comparison of predicted and ground-truth transient curves across three representative test scenarios (columns).

The reconstructed time series data matches the ground truth to a degree. The Encoder inputs contain more information, and their reconstruction results also match the ground truth. The Mapper distilled from the Encoder model, without the support of the time series data, shows a large prediction bias in some regions, but is generally in line with expectations. The LSTM and the fully connected network can predict accurately in most cases, but they can deviate severely from the actual curves in some cases. This indicates that the LSTM and fully connected models struggle to learn the complex mapping from the initial operating condition parameters to the full accident transient response curve of NPPs. The MSE and R^2 of the models are shown in Table 8.

Table 8. The MSE and R^2 of models.

	Encoder–Decoder	Mapper–Decoder	LSTM	Dense
MSE	0.00040	0.00161	0.00235	0.00213
R^2	0.99546	0.98302	0.97560	0.97767

The Encoder–Decoder architectural model achieves the best ratings in both MSE and R^2 , proving the feasibility of the approach to extract the latent spatial features of the transient response curves of reactor accidents through the Encoder–Decoder structural

model. Meanwhile, the Mapper–Decoder model is only lower than the Encoder–Decoder. It indicates that during the knowledge distillation, Mapper has a decrease in the latent spatial mapping ability relative to Encoder, but it still outperforms LSTM and Dense. Overall, the introduction of Attention and latent space has improved the model’s performance in predicting the reactor accident.

Figure 7 shows the error distribution of each model on the test set. The fully connected network lags behind the other three models in the tests for the T_s , RL , and PCP parameters.

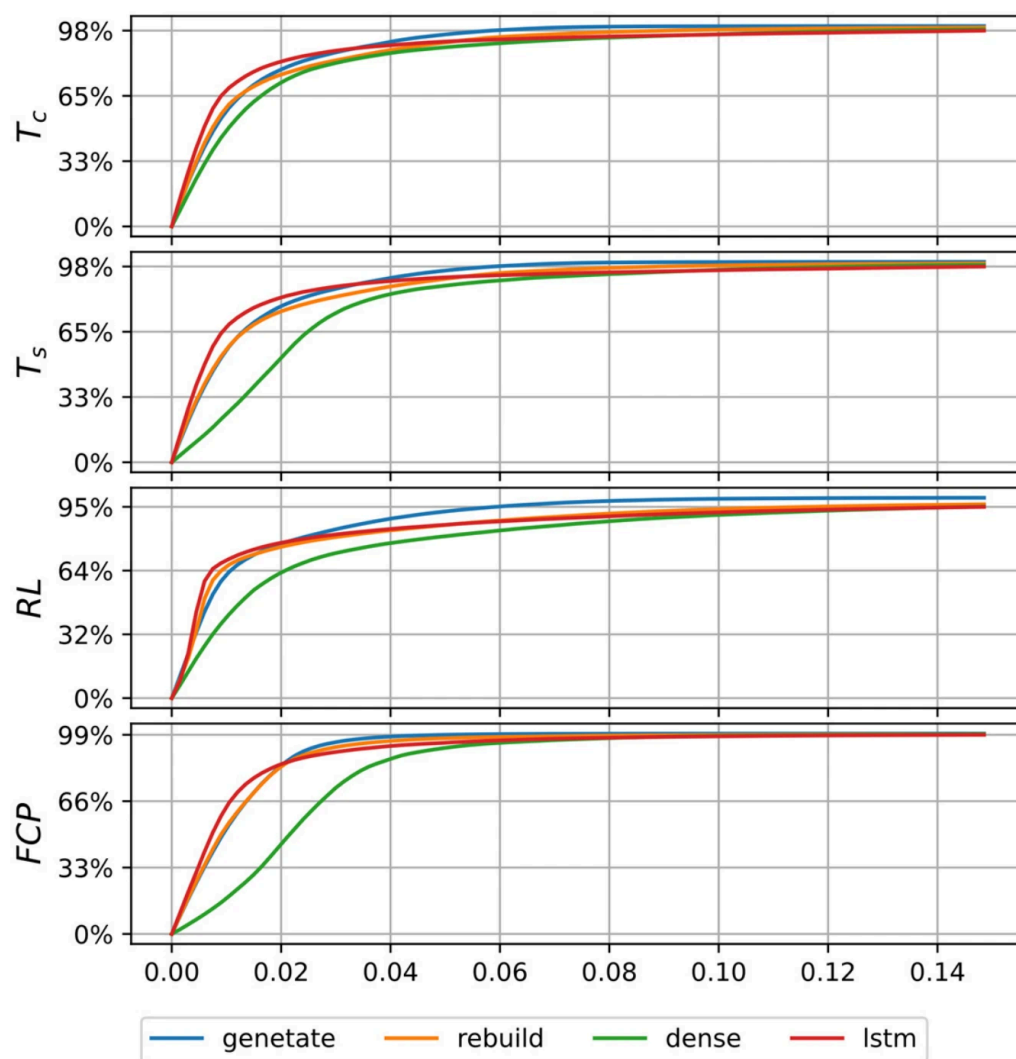


Figure 7. Comparative comparison of error distributions of different models.

In the comparison of LSTM with Decoder, Encoder, and Mapper, the distribution of LSTM within a small error range slightly outperforms the latter joint models. But the joint model of Encoder–Decoder and Mapper–Decoder outperforms LSTM when the error is large, suggesting that the distribution of the error of the joint model is more concentrated compared to that of LSTM.

A set of working condition data in the test set is selected and applied with noise levels of 10%, 15%, and 20% of the value to the condition data, in order to observe the performance of different models. As shown in Figure 8, the different models, after calculating the vectors with different noises added, corresponding to the results without added noise, all show some deviation. Among them, the deviation range of Encoder–Decoder is smaller, which means that the Encoder has higher resistance to noise, because the computation of Encoder–Decoder needs complete time series data, and the dependence on conditions is lower. In

addition, the floating range of the error is larger at the end of the accident. The model is more likely to follow a single pattern in the pre-accident response, with a lower dependence on the operating conditions, and a higher dependence on the operating conditions at the post-accident period.

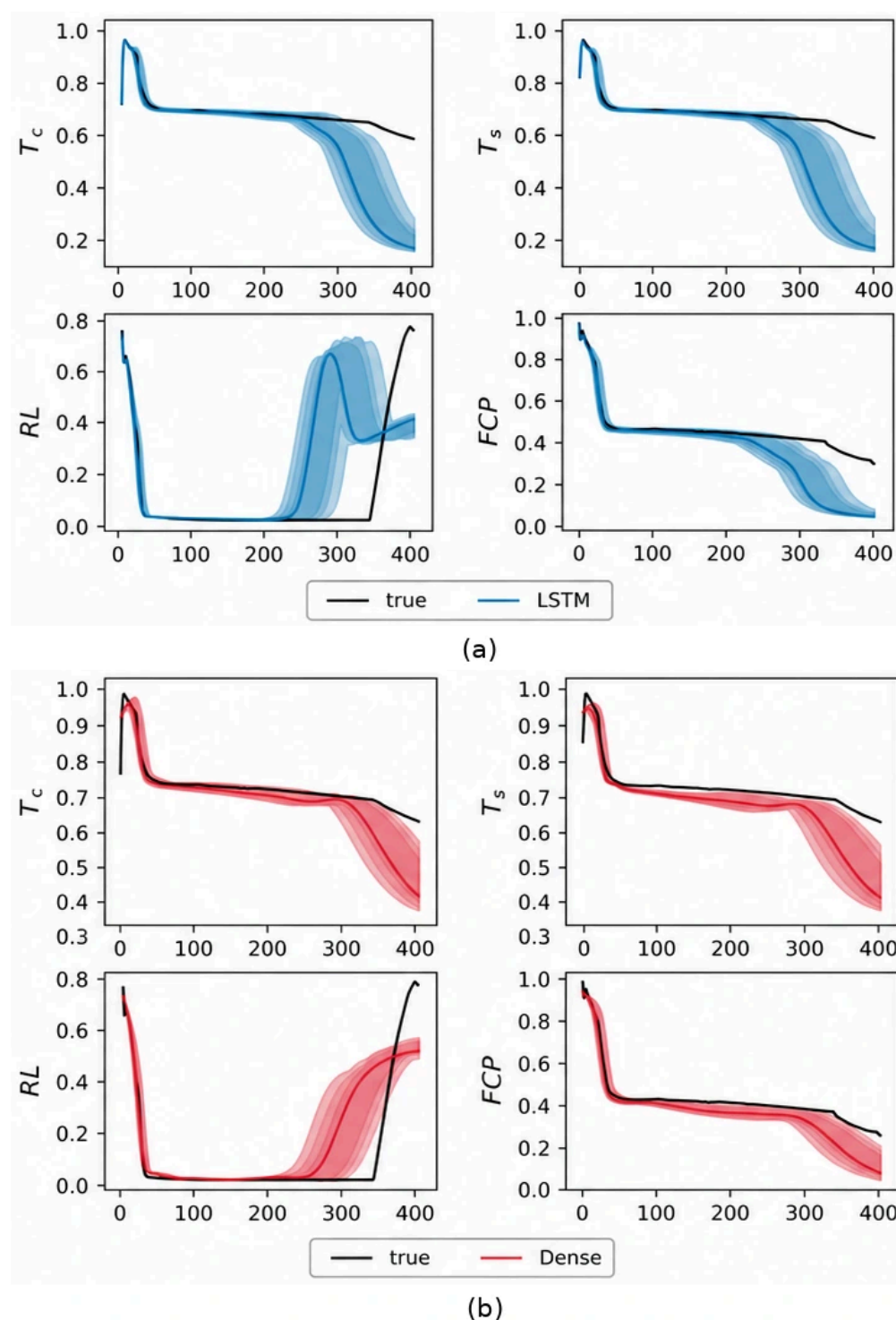


Figure 8. Cont.

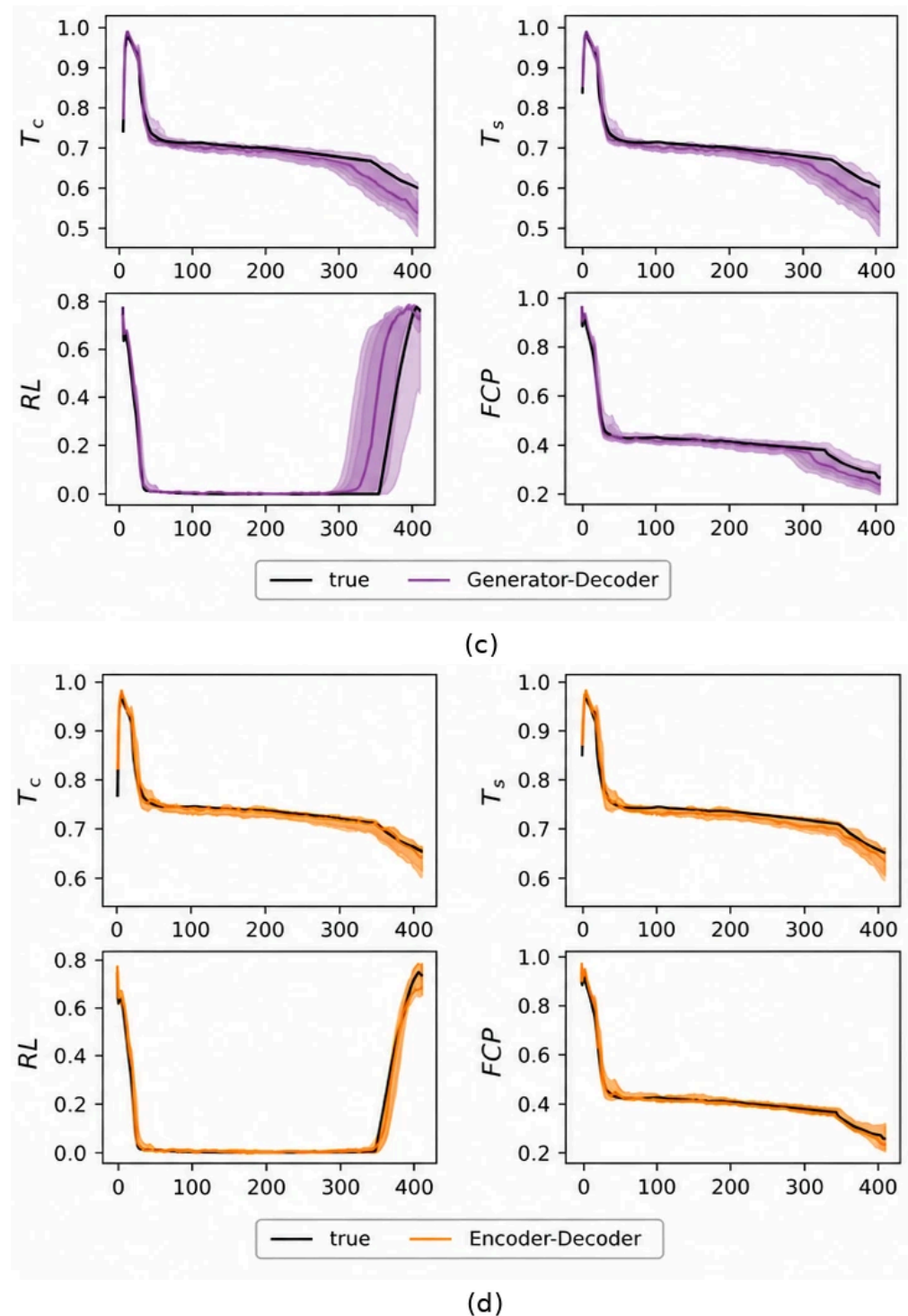


Figure 8. Comparison of model robustness after noise addition to input data. (a) LSTM; (b) Dense; (c) Mapper–Decoder; (d) Encoder–Decoder.

6. Conclusions and Future Outlook

6.1. Conclusions

In this study, an Attention mechanism is added on the basis of VAE in order to construct a transient response prediction model for nuclear power plant accidents. First, the VAE model with Attention added is used to construct the latent space through unsupervised training. Subsequently, knowledge distillation is used to extract the Mapper and construct a Mapper–Decoder model that predicts the accident transient response relying only on the operating condition parameters. The performance of the model is analyzed using a test set,

and the experimental results show that the prediction accuracy of the proposed model is higher than that of LSTM and Dense, and at the same time, it has strong robustness. The main conclusions are as follows:

- (1) It is feasible to construct a nuclear reactor prediction model with a generative neural network model, and the Attention mechanism brings an improvement in the generalization ability of the model and has good interpretability.
- (2) The proposed method abandons the traditional iterative sequential prediction approach and adopts one-step generation of full accident transients, which is verified to be feasible. However, the problem of high computation of the Attention mechanism brought about at the same time remains to be solved.
- (3) The study helps to provide a fast prediction model that can be used for sensitivity and uncertainty analysis of key safety parameters in nuclear power generation, optimal design of nuclear safety systems, nuclear safety analysis, and development of intelligent automatic control systems for nuclear power plants.

6.2. Methodological Challenges and Future Work

Although the proposed model demonstrates high accuracy in scaling prediction, it remains inherently a data-driven surrogate rather than a mechanistic solver. A Physics-Informed Neural Network (PINN) style residual against the two-fluid conservation equations was not adopted in the present work—because such a residual would penalize the model for learning the very numerical closure RELAP5 itself applies—and is left for follow-up work using stabilized formulations [22,23]. Unlike traditional thermal-hydraulic codes such as RELAP5, the model does not strictly guarantee real-time thermodynamic conservation and relies heavily on static operational snapshots. Consequently, while it serves as an efficient acceleration tool for rapid screening, it cannot fully replace rigorous thermal-hydraulic codes for formal nuclear safety licensing at this stage. The current output channels capture only the system-level thermal-hydraulic state; coupled safety criteria such as the Departure from Nucleate Boiling ratio (DNBR) and critical heat flux (CHF) proximity require channel-resolved local conditions and are not directly computable from the present surrogate.

To justify the computational efficiency of the framework, a comprehensive complexity analysis is performed. As summarized in Table 9, our model significantly reduces memory footprint and inference time compared to baseline architectures, achieving a promising trade-off between accuracy and deployment cost. However, the current feedforward architecture lacks a dynamic feedback loop, which may limit its robustness under extreme out-of-distribution (OOD) transient scenarios. Future efforts will focus on implementing closed-loop correction mechanisms and standardized stress-testing protocols to enhance edge-case reliability.

Table 9. Analytical model-complexity comparison.

Model	Parameter Count ($\approx$)	Activation Memory Profile	Dominant Operational Complexity ($\mathcal{O}$)	Relative Inference Cost
Encoder–Decoder (This Work)	$\sim 3.2 \times 10^5$	High (full sequence in/out)	$\mathcal{O}(L^2 \cdot d)_{Self-Attention}$ ($L = 500, d = 64$)	$\approx 2.8\times$
Mapper–Decoder (This Work)	$\sim 2.1 \times 10^5$	Moderate (Mapper: static vector; Decoder: full sequence)	$\mathcal{O}(L^2 \cdot d)_{Self-Attention}$ ($L = 500, d = 64$)	$1.0\times$ (Baseline)

Table 9. Cont.

Model	Parameter Count ($\approx$)	Activation Memory Profile	Dominant Operational Complexity ($\mathcal{O}$)	Relative Inference Cost
LSTM Baseline	$\sim 1.0 \times 10^5$	Low Moderate (Sequential)	$\mathcal{O}(L \cdot 4(d_{\text{in}} + d_{\text{out}}) \cdot d)$ <i>autoregressive</i>	$\approx 1.3 \times (500 \text{ sequential loops})$
Dense Baseline	$\sim 3.3 \times 10^4$	Very low (Per individual timestep)	$\mathcal{O}(d_{\text{in}} \cdot d_{\text{out}}) \times L \text{ calls}$	$\approx 0.6 \times (\text{per-step}, L, \text{calls})$
Standard VAE (Matched Depth, No Attention)	$\sim 1.8 \times 10^5$	Moderate	$\mathcal{O}(L \cdot d^2) \text{ feedforward}$	$\approx 0.9 \times$

Translating the trained model into real-world nuclear power plant (NPP) applications introduces an inevitable “Sim-to-Real” gap. From an engineering perspective, the input features can be categorized into three classes based on observability: Class A (directly measurable physical quantities, e.g., pressure, temperature), Class B (derivable parameters via high-fidelity observers), and Class C (unobservable structural degradation states). Addressing the stochastic uncertainty of Class C features and maintaining latent space continuity under sparse data regimes remain critical open challenges for future field deployment.

Author Contributions: Conceptualization, B.P.; methodology, Y.L., Z.P. and G.C.; software, G.Q.; validation, S.Z.; formal analysis, D.C. and G.C.; investigation, Q.A.; resources, G.Q., S.Z., Y.Z., D.C. and G.C.; data curation, B.P., Y.L., S.Z. and Y.Z.; writing—original draft preparation, B.P., Y.L., G.Q., S.Z., Z.P. and Y.Z.; writing—review and editing, B.P., Y.L., D.C., G.C. and Q.A.; visualization, Y.Z. and D.C.; supervision, Z.P., D.C. and Q.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to security restrictions related to nuclear facility data.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tian, D.; Deng, J.; Vinod, G.; Santhosh, T.V.; Tawfik, H. A constraint-based genetic algorithm for optimizing neural network architectures for detection of loss of coolant accidents of nuclear power plants. *Neurocomputing* **2018**, *322*, 102–119. [\[CrossRef\]](#)
2. Wang, M.-D.; Lin, T.-H.; Jhan, K.-C.; Wu, S.-C. Abnormal event detection, identification and isolation in nuclear power plants using LSTM networks. *Prog. Nucl. Energy* **2021**, *140*, 103928. [\[CrossRef\]](#)
3. Zhang, J.; Wang, X.; Zhao, C.; Bai, W.; Shen, J.; Li, Y.; Pan, Z.; Duan, Y. Application of cost-sensitive LSTM in water level prediction for nuclear reactor pressurizer. *Nucl. Eng. Technol.* **2020**, *52*, 1429–1435. [\[CrossRef\]](#)
4. Ramezani, I.; Vosoughi, N.; Ghofrani, M.B. Application of deep learning techniques for nuclear power plant transient identification. *Ann. Nucl. Energy* **2023**, *194*, 110113. [\[CrossRef\]](#)
5. Nicolau, A.d.S.; Cabral Pinheiro, V.H.; Schirru, R.; da Silva, M.d.C.; de Mello, A.S.; de Lima, A.M.M. Deep neural networks for estimation of temperature values for thermal ageing evaluation of nuclear power plant equipment. *Prog. Nucl. Energy* **2023**, *156*, 104542. [\[CrossRef\]](#)
6. Kim, H.; Arigi, A.M.; Kim, J. Development of a diagnostic algorithm for abnormal situations using long short-term memory and variational autoencoder. *Ann. Nucl. Energy* **2021**, *153*, 108077. [\[CrossRef\]](#)
7. Choi, Y.J.; Kim, H.K.; Baek, W.P.; Chang, S.H. Hybrid accident simulation methodology using artificial neural networks for nuclear power plants. *Inf. Sci.* **2004**, *160*, 207–224. [\[CrossRef\]](#)
8. Montes, J.L.; François, J.L.; Ortiz, J.J.; Martín-del-Campo, C.; Perusquía, R. Local power peaking factor estimation in nuclear fuel by artificial neural networks. *Ann. Nucl. Energy* **2009**, *36*, 121–130. [\[CrossRef\]](#)

9. Antonello, F.; Buongiorno, J.; Zio, E. Physics informed neural networks for surrogate modeling of accidental scenarios in nuclear power plants. *Nucl. Eng. Technol.* **2023**, *55*, 3409–3416. [\[CrossRef\]](#)
10. Saghafi, M.; Ghofrani, M.B. Real-time estimation of break sizes during LOCA in nuclear power plants using NARX neural network. *Nucl. Eng. Technol.* **2019**, *51*, 702–708. [\[CrossRef\]](#)
11. Li, X.; Huang, T.; Cheng, K.; Qiu, Z.; Sichao, T. Research on anomaly detection method of nuclear power plant operation state based on unsupervised deep generative model. *Ann. Nucl. Energy* **2022**, *167*, 108785. [\[CrossRef\]](#)
12. Li, X.; Cheng, K.; Huang, T.; Tan, S. Research on false alarm detection algorithm of nuclear power system based on BERT-SAE-iForest combined algorithm. *Ann. Nucl. Energy* **2022**, *170*, 108985. [\[CrossRef\]](#)
13. Saraswat, S.P.; Munshi, P.; Allison, C. Characteristics and linear stability analysis of RELAP5 two-fluid model for two-component, two-phase flow. *Ann. Nucl. Energy* **2021**, *151*, 107948. [\[CrossRef\]](#)
14. Estévez-Albujá, S.; Jiménez, G.; Vázquez-Rodríguez, C. AP1000 IRWST numerical analysis with GOTHIC. *Nucl. Eng. Des.* **2021**, *372*, 110991. [\[CrossRef\]](#)
15. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Scabini, L.F.S.; Bruno, O.M. Structure and performance of fully connected neural networks: Emerging complex network properties. *Phys. A Stat. Mech. Its Appl.* **2023**, *615*, 128585. [\[CrossRef\]](#)
17. Liao, B.; Chen, S.; Wang, X.; Cheng, T.; Zhang, Q.; Liu, W.; Huang, C. MapTR: Structured Modeling and Learning for Online Vectorized HD Map Construction. *arXiv* **2022**, arXiv:2208.14437.
18. Chen, S.; Guo, W. Auto-Encoders in Deep Learning—A Review with New Perspectives. *Mathematics* **2023**, *11*, 1777. [\[CrossRef\]](#)
19. Yang, Z.L.; Zhang, S.Y.; Hu, Y.T.; Hu, Z.W.; Huang, Y.F. VAE-Stega: Linguistic Steganography Based on Variational Auto-Encoder. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 880–895. [\[CrossRef\]](#)
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
21. Wei, X.; Zhang, L.; Yang, H.-Q.; Zhang, L.; Yao, Y.-P. Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks. *Geosci. Front.* **2021**, *12*, 453–467. [\[CrossRef\]](#)
22. Ye, L.; Li, Z.-H.; Yan, H.-N.; Liu, C.-L.; Cho, H.H.; Guo, T. Predicting the Film-Cooling Effectiveness of Compound-Angle Holes Using a POD-Based Hybrid Deep Learning Model. *Aerosp. Sci. Technol.* **2026**, *176*, 112590. [\[CrossRef\]](#)
23. Yang, T.; Qian, Z.; Hang, N.; Liu, M. S-PINN: Stabilized physics-informed neural networks for alleviating barriers between multi-level co-optimization. *Comput. Methods Appl. Mech. Eng.* **2025**, *447*, 118348. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.