

Research Article

Application of Deep Neural Network to an Accelerated Prediction of a Severe Accident in Nuclear Power Plants

Semin Joo ¹, Yeonha Lee,¹ Seok Ho Song,¹ Kyusang Song,² Mi Ro Seo,² Sung Joong Kim ³, and Jeong Ik Lee ¹

¹Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291, Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

²Korea Hydro and Nuclear Power Central Research Institute, 70, Yuseong-daero 1312beon-gil, Yuseong-gu, Daejeon 34101, Republic of Korea

³Department of Nuclear Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul 04763, Republic of Korea

Correspondence should be addressed to Jeong Ik Lee; jeongiklee@kaist.ac.kr

Received 3 September 2024; Accepted 29 May 2025

Academic Editor: Benkun Qi

Copyright © 2025 Semin Joo et al. International Journal of Energy Research published by John Wiley & Sons Ltd. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Recent nuclear severe accidents have spurred interest in the development of advanced accident management support tools (AMSTs) to enhance decision-making during crises. This study examines the efficacy of deep neural networks (DNNs) in accelerating severe accident predictions within nuclear power plants (NPPs), focusing on a loss-of-component-cooling-water (LOCCW) accident scenario. Through analysis of 10,780 simulated LOCCW accident scenarios across varied component failures and mitigation strategy implementations, time series datasets were synthesized at 15, 30, and 60-min intervals. The evaluation demonstrated that convolutional neural network (CNN)-integrated models outperformed standalone architectures in prediction accuracy across all temporal resolutions. Notably, higher temporal resolutions in training datasets significantly improved mean absolute error (MAE) and root mean squared error (RMSE), thereby enhancing prediction precision for immediate subsequent time steps. However, the augmentation of temporal resolution did not uniformly improve overall scenario prediction performance, as assessed by dynamic time warping (DTW) distance, due to cumulative prediction error in higher resolution models. These findings elucidate the nuanced relationship between temporal resolution and predictive accuracy, offering valuable insights for the development of sophisticated AMSTs aimed at bolstering nuclear safety and accident management strategies.

1. Introduction

The prevention and mitigation of a severe accident is one of the biggest challenges in nuclear safety [1]. A severe accident is an accident condition of a nuclear power plant (NPP) involving serious core degradation that could bring about off-site radiological consequences. From the past three severe accidents, namely the Three Mile Island (TMI), Chernobyl, and Fukushima nuclear accidents, the lesson learned is that reducing human error is crucial for preventing and mitigating severe accidents [2]. Since then, the nuclear industry has adopted various safety measures to reduce human error, such as passive safety systems, updated emergency operating

procedures, and a mandated submission of severe accident management guidelines (SAMGs) [3]. However, when an operator is placed in an accident situation, it is extremely difficult for the operator to make a sound judgment. Any misjudgment made by the operator can aggravate the accident, as observed in the TMI-2 nuclear accident.

There has been a growing interest in the nuclear industry to develop accident management support tools (AMSTs) that support the decision-making of operators to reduce their mental workload in accident situations [4]. Machine learning (ML) has reared up as a powerful tool that has a strong generalization ability and robustness to noisy data [5]. Hence, attempts have been made to apply ML to the field of nuclear

safety and developing AMSTs. Antonello et al. [6] developed a physics-informed neural network to model a loss-of-heat-sink scenario in a high-temperature heat pipe reactor. By integrating the reactivity feedback equation into the loss function, the developed model provided accurate solutions and physics-consistent results. Gu et al. [7] proposed a sequence-to-sequence (Seq2Seq)-based long short-term memory (LSTM) model to predict safety-critical parameters and their future trends. Under four typical NPP accidents (steam generator [SG] tube rupture [SGTR], loss of coolant accident [LOCA], control rod ejection accident, and containment steam pipe rupture), a total of 92 safety parameters were selected as target parameters. The proposed model could successfully estimate the trends of key system parameters under the influence of operator action factors in multiple abnormal operating conditions. Another study proposed an AI-based system to predict the remaining trip time (RTT) in NPPs [8]. The system is designed to assist operators in managing abnormal states by providing information on the remaining time until a reactor shutdown occurs. The algorithm incorporates a diagnosis function that identifies scenarios for abnormal states of NPPs and uses a light gradient boosting machine for RTT prediction. Nguyen and Diab [9] developed a multivariate time-series ML model to predict the transient response of an NPP undergoing an SGTR. The model employs recurrent neural networks (RNNs), including the LSTM, gated recurrent unit (GRU), and a hybrid convolutional neural network (CNN)-LSTM model. To address the uncertainty inherent in such predictions, a Bayesian neural network (BNN) was implemented. It was concluded that BNN can serve as a valuable tool for modeling uncertainty of the ML model, particularly in safety-critical domains such as nuclear accident analysis. Tiep et al. [10] utilized data assimilation (DA) and deep neural networks (DNNs) to enhance the prediction of reflood tests. By integrating DNN, the model could accurately predict high flooding rate scenarios compared to the stand-alone DA method with high computational efficiency.

In this light, the author's research group has previously investigated the applicability of ML to a broader range of severe accident scenarios of an NPP. In this study, a versatile surrogate model using DNNs and the rolling window forecast method to predict a loss-of-component-cooling-water (LOCCW) accident in an NPP was developed [11]. A rolling window forecast method is one of the classic time series forecasting techniques where a fixed-size window moves through the sequential data one step at a time. At each step, a model is trained on the data within the window, predicting the next time point. In this study, the surrogate model was trained with accident datasets that represent the thermal-hydraulic (TH) behavior of an OPR1000 (a Korean standard NPP) reactor under multi-component failures occurring at various times. It was discovered that the proposed surrogate models can make accurate predictions of the accident scenarios within 2 s, which is significantly faster than the conventional severe accident analysis codes such as modular accident analysis program (MAAP [12]). Hence, it was concluded that the proposed surrogate model could serve as a severe accident simulation tool

for AMST. In this work, the time series data was produced at an interval of 1 h for a 72-h accident scenario. However, as the plant states can change dynamically in an accident situation, hourly-basis data may not sufficiently capture the plant dynamics. Thus, a time series dataset with a narrower time resolution may be needed to train the surrogate models.

There have been several studies investigating the effect of time resolution on the performance of time series forecasting models. Kim and Cho [13] predicted the residential energy consumption using CNN-LSTM architecture based on a rolling window forecast method. The residential data was processed into diverse time intervals, ranging from minutes to weeks, to train the proposed CNN-LSTM network. It was observed that the error in predicting the residential energy consumption increased as the time resolution increased (i.e., as the time interval became narrower). Another study on predicting residential energy consumption was conducted with a multi-headed attention-based deep learning algorithm [14]. They have tested various supervised learning techniques within various time resolutions to discover that the forecasting error and the time resolution of the training data do not necessarily have a succinct linear relationship. Both studies indicate a counterintuitive result that the time series forecasting model may not necessarily be enhanced, albeit the time resolution is increased. From this point, the same question arises for the forecasting of a nuclear accident progression using AI: Does the increase in time resolution of the training data have a positive effect on the accurate prediction?

In this background, this study investigates the sensitivity of the surrogate model's performance to time resolution. In the previous work [11], a surrogate model that predicts the progression of a LOCCW accident on an hourly basis has been developed. The surrogate model was constructed based on DNNs consisting of CNN and LSTM layers. As an expansion of the previous study, the surrogate models are trained at various time resolutions—60, 30, and 15 min. To consider the hysteresis effect of the TH variables, the surrogate model will use the information from the previous three time steps. Also, various types of DNN architectures other than CNN and LSTM will be employed to investigate the suitable architecture for severe accident forecasting. The main contributions of this study are summarized below.

- In a dynamic accident situation, the prediction time resolution should be as narrow as possible. This study explicitly investigates the effect of narrowing the time resolution on the prediction accuracy and its related concerns, thus providing insights in developing a practical AMST.
- From the detailed sensitivity study on the time resolution identifies the specific TH variables that are especially sensitive to the accumulation of calculation errors. From these findings, the special considerations when developing an AMST may be deduced.
- By comparing the performances of various DNN architectures, the characteristics of each architecture in

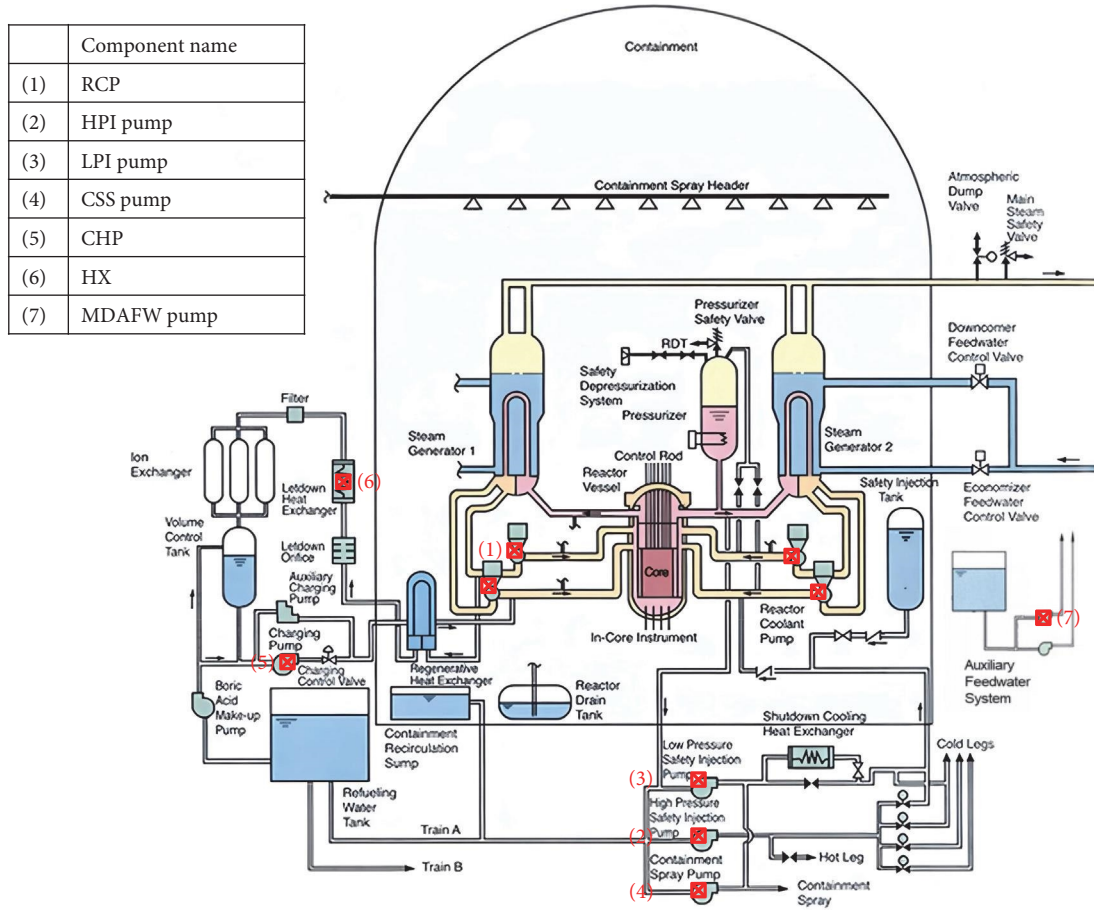


FIGURE 1: Locations of possible component failures in the OPR1000 system.

predicting a complex multivariable time series can be investigated. Thus, the optimal DNN architecture for developing an AMST can be recommended.

The remainder of this paper is organized as follows: Section 2 describes the selected accident scenario and how the data is generated. In Section 3, the training methodologies, such as the architectures of the proposed models, validation strategy, and rolling window forecast method, are explained. The training results are presented in Section 4, based on two performance indicators. Section 5 summarizes the results and proposes the limitations and future works of this study.

2. Data Preparation

2.1. Targeted Accident Scenario. The reference plant that this study aims to simulate is Optimized Power Reactor 1000 (OPR1000), a Korean standard nuclear plant operated by Korea Hydro & Nuclear Power (KHNP). The baseline accident scenario is a total-loss-of-component-cooling-water (TLOCCW) accident occurring at an OPR1000 plant. The TLOCCW accident scenario was selected by referring to the Level 2 Probabilistic Safety Assessment (PSA) report, which concluded that the TLOCCW accident has the highest plant damage frequency among the various severe accident scenarios. In a TLOCCW accident, all

branches of the primary component-cooling water system fail initially, and the main coolant pump trips. Pressurizer spray and heaters simultaneously fail, eliminating pressure control. Makeup flow is isolated, primary pressure rises, and the reactor scrams. After shutdown, decay heat is removed solely through secondary-side cooling. All turbine-driven auxiliary feedwater pumps are assumed inoperable; only the motor-driven pump supplies feedwater. The automatic steam-relief mode is disabled to directly observe the effect of deliberate reactor coolant system (RCS) depressurization. Subsequent reactor coolant pump (RCP) seal failure causes coolant leakage.

Unlike the TLOCCW event tree presented in the OPR1000 PSA report, this study assumes that the cooling components fail at random times rather than all at once at the outset. Seven safety components were selected to fail at random times: (1) RCP, (2) high-pressure injection (HPI) pump, (3) low-pressure injection (LPI) pump, (4) containment spray system (CSS) pump, (5) charging pump (CHP), (6) letdown heat exchanger (HX), and (7) motor-driven auxiliary feedwater (MDAFW) pump (Figure 1). This is to confirm the versatility of the surrogate model. For simplicity, such type of accident scenario will be denoted as a LOCCW accident for the rest of this paper.

Additionally, the information about whether a component has failed is expressed in binary form. SAMG is a

TABLE 1: Severe accident management strategies for OPR1000 [15].

SAM number	Strategy	Objective	Parameter for the entry condition
SAM-1	SG external injection	The secondary side of the SG serves as the heat sink for the nuclear heat. By injecting water into the SG, the RCS is depressurized, facilitating the injection of coolant into the RPV.	SG water level
SAM-2	RCS depressurization	If the RPV fails at a high-pressure state, there is a possibility of a direct containment heating (DCH) phenomenon [16]. Thus, it is critical to depressurize the RCS by opening the safety depressurization system valve. By reducing the primary system pressure, RCS injection (which is the next SAM strategy) is facilitated.	Primary system pressure
SAM-3	RCS external injection	By injecting water into the RPV, the core can be cooled down, preventing the failure of the RPV.	Reactor vessel water level

guideline that describes a set of strategies and steps that are recommended to be taken to mitigate the progression of a severe accident. In response to the amendment of the Nuclear Safety Act in 2016, KHNP developed eight strategies for the SAMG. Once the NPP meets the severe accident management (SAM) entry condition, that is, core exit temperature exceeds 650°C, the SAM strategies are implemented sequentially by the operator. Among them, the first three strategies, which are dedicated to in-vessel cooling, were selected (Table 1). The strategies are allowed to be activated when a certain parameter meets a predetermined condition. For example, the SG external injection is implemented when the SG water level is below a certain elevation. To confirm the versatility of the surrogate models, the three strategies are assumed to be initiated randomly throughout the 72-h accident scenario.

Here, the detailed sampling method for determining the seven component failure times and three SAM strategy activation times is explained. For both component failure and SAM activation, it must be determined first whether the component/SAM strategy will fail/activate or not during the 72-h accident. Each failure/activation was assumed to occur throughout the accident with a uniform probability of 1/2. Once a component has been determined to fail, the timing of the failure was sampled from $t = 1$ h to $t = 72$ h, with a 1-h interval, following a uniform random distribution. However, the timing of RCP seal LOCA was an exception—it follows a log-normal probability distribution function with its average centered at $t = 1$ h. 99.5% of the whole dataset, the RCP seal LOCA occurs 1 h after the accident initiation, meaning that the accident starts with RCP seal LOCA followed by random breakdowns of other six components.

Also, there were exceptions in the SAMG activation timings. Even if a specific SAM activation time (t_{SAM}) is stipulated in the MAAP input file, the SAM strategy is not necessarily turned on at $t = t_{\text{SAM}}$. This is because the SAMG allows the activation of SAM strategies when a certain thermal hydraulic condition is satisfied. Take SAM-1 for an example—assume that the activation time of SAM-1 was stated as $t = 5$ h. However, if the SG water level at $t = 5$ h is above the SAM-1 entry condition, the SAM-1 strategy cannot be switched on at $t = 5$ h. In such cases, the activation

TABLE 2: List of TH variables used as the input/output parameters of the surrogate model.

No.	TH variables
1	Primary system pressure (PPS)
2	Cold leg temperature
3	Hot leg temperature
4	Reactor vessel water level (RVWL)
5	Steam generator pressure (SGP)
6	Steam generator water level (SGWL)
7	Maximum core exit temperature (Max CET)
8	Containment pressure (CTMTP)
9	Pressurizer pressure (PZRP)
10	Pressurizer water level (PZRWL)

times of each SAM strategy must be corrected based on the log files of MAAP output. Such a correction process leaves out the accident scenarios in which all component failure times and SAM strategy activation times coincide with each other. In the end, a total of 10,780 accident scenarios could be generated and simulated with MAAP code.

2.2. Selection of Thermal Hydraulic Variables. The proposed surrogate model predicts the plant state using a small number of thermal hydraulic (TH) variables. Among various plant parameters, 10 parameters have been chosen as the input/output parameters of the surrogate models (Table 2). It should be noted that the ten TH variables were chosen solely for their reliable observability from the main control room (MCR) during a severe accident, not through statistical methods like correlation analysis or dimensionality reduction. Even under extreme temperature, pressure, and radiation, they remain dependable for operators and the technical support center. Indicators such as hydrogen or fission-product concentrations were excluded because they fall outside the thermal-hydraulic domain.

In the previous study [11], only seven TH variables (Nos. 1–7) have been utilized for training the surrogate model. However, the findings from recent studies [17, 18] have revealed that the accuracy of predicting the RVWL can be improved by including additional TH variables. Thus, three

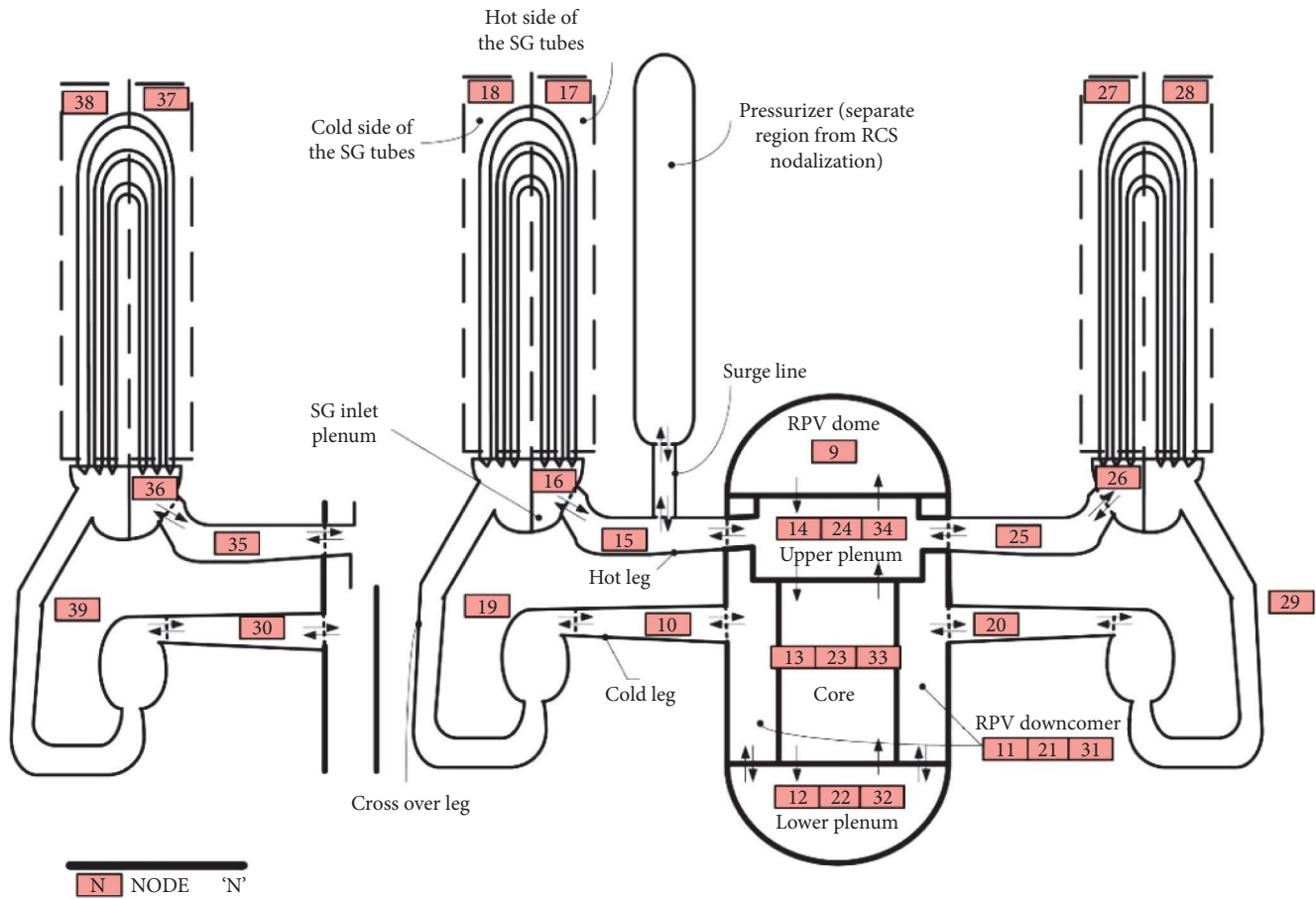


FIGURE 2: MAAP5 nodalization of the primary system.

TH variables were added to the training dataset, listed as Nos. 8–10 in Table 2.

2.3. Generation of MAAP Data. The MAAP calculation was implemented under the nodalization scheme depicted in Figure 2, showing the water nodes for the OPR1000 NPP. The two SGs are connected through four intermediate legs, at which the RCP seal LOCA is assumed to occur. The node numbers are 19, 29, 39, and 49, each of which is located right next to the RCPs. The rupture size of the RCP seal LOCA was assumed to be constant.

With predetermined component failure times and SAMG activation times, an accident scenario can be generated. Such predetermined information is written as an input file format for MAAP code. A total of 10,780 accident scenarios were considered and put into the MAAP code one by one. Then, the MAAP code calculates a wide range of TH variables as a time series whose length is 72 h with an interval of 5 min. Among the TH variables, the variables that have been selected in Section 2.2. were extracted from the output files and were processed to form a training dataset. The original MAAP datasets were processed into time series datasets whose length was 72 h with an interval of 60, 30, and 15 min. This eventually generated a total of 786,940, 1,573,880, and 3,147,760 data pairs, respectively. The

detailed processing methods and training methodology are discussed in the next section.

2.4. Validation of the MAAP Data. MAAP is a widely used computer code for simulating and analyzing severe accidents in NPPs. Several efforts have been made to validate the accuracy of the MAAP code using real severe accident data, such as the TMI-2 accident and the Fukushima-Daiichi accident. For example, a mock-up experiment simulating the TMI-2 accident was compared with the results calculated using the MAAP code [19]. In this analysis, the primary system pressures and pressurizer water levels showed good agreement during most of the simulated period. In another study using MAAP code to simulate the Fukushima-Daiichi accident, the MAAP analysis successfully reproduced the behavior of the plants before the core melt of the reactor unit 1 [20].

However, it should be noted that the NPP types in the TMI-2 and Fukushima-Daiichi accidents were Babcock & Wilcox's pressurized water reactor and General Electric's boiling water reactor, which are different from the target NPP type in this study (KHNP's OPR1000). Since no OPR1000-type NPPs have experienced severe accidents throughout their operation history, the validation of MAAP code with real severe accident data can only be done on reactor types other than OPR1000. Nevertheless, the major physical phenomena governing severe

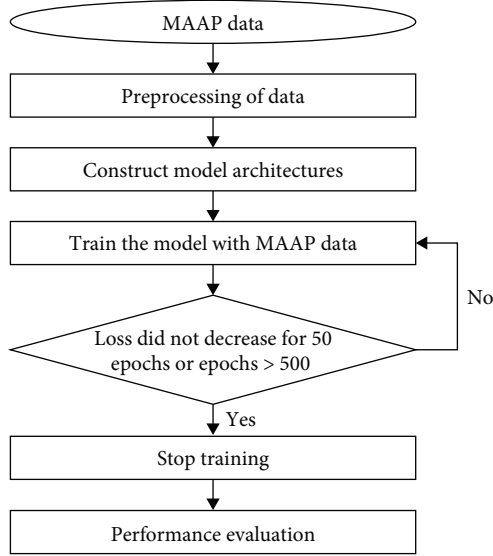


FIGURE 3: Surrogate model training and evaluation procedure.

accidents are expected to be similar, and therefore the data generated from MAAP calculations can have high confidence. Furthermore, the reference input used in the MAAP calculation was developed by the utility company operating OPR1000, and therefore the input and output are all qualified to generate data with high confidence.

3. Training Methodology

Using the accident dataset obtained from MAAP calculations, various surrogate models were developed. The process for training and evaluating the surrogate models is illustrated in Figure 3.

3.1. Data Preprocessing. First, the MAAP data is processed into a form that is appropriate for training. As the selected TH variables have different scales and ranges, the values need to be normalized into the same scale. Every feature is transformed into a value ranging between 0 and 1.

Also, the information about whether a component has failed or not is expressed in a binary form. For example, if the HPI pump has failed at a certain time step, the failure state for the HPI pump is “1”. The same scheme applies to the expression of the activation of a SAM strategy. If a certain SAM strategy is activated, the activation state for that SAM strategy is 1. The information about a specific time step is composed of 20 elements—10 TH variables that have been min–max normalized, 7 component failure states in binary values, and 3 SAM strategy activation states in binary values.

Although the overall training process is not distinct from other machine learning processes, the input data structure itself is unique. In other studies, related to the application of ML on the prediction of nuclear accidents, the input data are usually limited to the information dedicated to a single accident scenario. For instance, Radaideh et al. [21] used the break size and other 40 safety parameters to predict the critical safety parameters (e.g., core pressure, break flow rate) of a LOCA scenario. However, in a severe accident

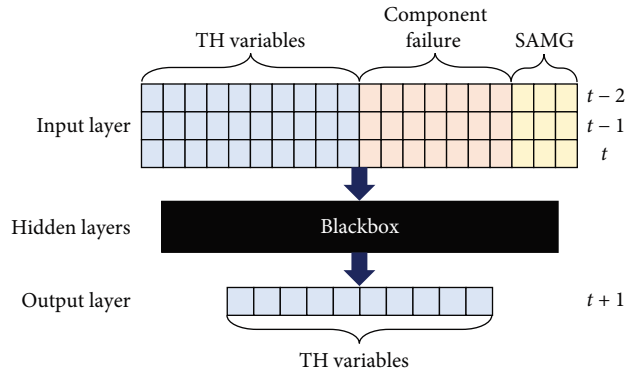


FIGURE 4: Structure of the DNN model.

situation, it is difficult to identify the type (e.g., LOCA) and details (e.g., break size) of the accident. The operators rely on the parameters observed in the MCR and whether a certain component has failed or not. Hence, this study gains novelty and practicality in the way of reflecting the plant’s status on the input data.

To account for hysteresis, the surrogate model proposed in this study forecasts the plant state at the next time step using the information of the past three time steps. In the previous study [11], the impact of the number of time steps was investigated by quantitatively evaluating the autocorrelation of each TH variable, and the highest autocorrelation was found for up to three previous time steps. When comparing prediction results between models utilizing one and three time steps, the three-time-step model showed lower average errors. Accordingly, three sets of information are organized into a 3×20 matrix form so that they are subsequently put into the input layer of the DNN models. The structure of the input layer is depicted in Figure 4.

The last step in data pre-processing is splitting the whole dataset into a training set, validation set, and test set. The dataset is divided into (training):(validation):(test) = 7:2:1. The details of the validation scheme will be explained in Section 3.3.

3.2. DNN Structure. Before training the neural network models with the prepared data, the DNN structure should be configured. While all models follow the structure illustrated in Figure 4, the types of hidden layers are varied. Five DNN structures have been tested in this study, using the layers that are famous for forecasting multivariable time series. CNNs and RNNs have been tested both independently and separately. The detailed structure of the proposed architectures will be discussed in the following subsections.

It should be noted that the selection of the hyperparameters has been determined empirically. The hyperparameters, such as the number of filters in the convolutional layers or the number of hidden sizes in the RNN layers, were increased from 50 to 400. Figure 5 shows that the root mean squared error (RMSE) of the five models decreases continuously as the number of filters/hidden sizes increases. Hence, the number of filters/hidden sizes of the DNN models is fixed at 400 for the rest of this study. Although there is a possibility that the model’s prediction accuracy can be increased by

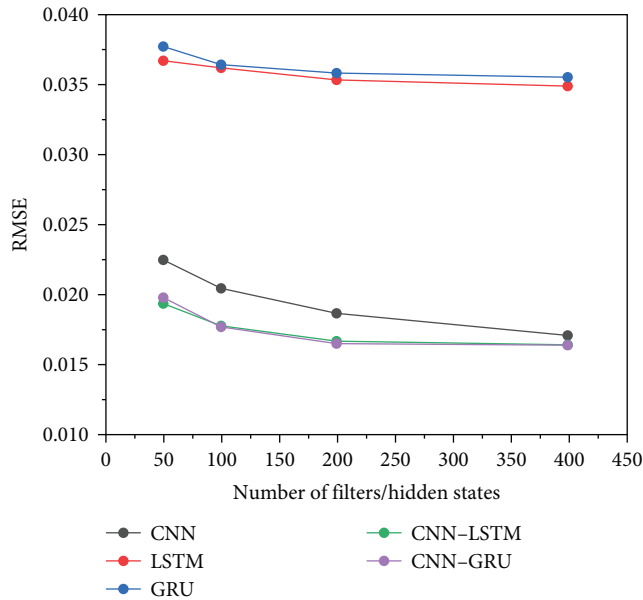


FIGURE 5: RMSE vs. number of filters/hidden size of the DNN models.

further enlarging the number of filters/hidden sizes, the number has been limited to 400 for training efficiency.

Also, all training and test processes have been fulfilled by the Keras Application Programming Interface (API) of the TensorFlow platform.

3.2.1. CNN. CNN is one of the popular DNN architectures that imitates a human optic nerve system [22]. Using convolution operators, the spatio-temporal information of the input data can be retained while dramatically reducing the amount of computation. Generally, CNNs are composed of convolutional layers, pooling layers, and fully connected layers. At the convolutional layer, a convolution operation is taken between a kernel of a fixed size and the input matrix based on a sliding window method.

To construct the hidden layers of a CNN model, a one-dimensional convolutional layer is connected to a fully connected layer. The convolutional layer is composed of 400 filters whose dimension is 3×3 , and the subsequent fully connected layer is also composed of 400 fully connected nodes. Both layers are outputted via the ReLU activation function. The results of the fully connected layer are passed to the output layer, which is also a dense layer activated by a sigmoid function.

3.2.2. LSTM. LSTM is a type of RNN. It is widely recognized that the RNN successively transfers the hidden state at the previous time step to the next time step. However, if the length of the sequential data is too long, the information at the initial stages cannot be sufficiently delivered to the back-stages. LSTM was developed to resolve this long-term dependency problem, utilizing its unique gate structures [23].

LSTM is a popular method used in time series forecasting and natural language processing due to its ability to retain long-term memory. As this study aims to develop a surrogate

model that forecasts a multivariate time series, the LSTM network was also considered as a candidate architecture. To construct an LSTM-based surrogate model, a stacked LSTM architecture was considered. A stacked LSTM network is composed of multiple consecutive LSTM layers, which enhances the model's ability to capture the complex and long-term dependencies of the incoming multivariate time series. In this study, two identical LSTM layers were connected, each of which consisted of 400 units and activated by a hyperbolic tangent function. The two LSTM layers can be connected by passing all hidden state information at the preceding layer to the next layer.

It should be noted that the dropout scheme was not applied at any of the architectures that will be further discussed. The dropout scheme intentionally drops out a small portion of the training dataset to avoid overfitting. However, this study assumes that dividing the training and validation set to separately track the loss values from the validation set is enough to prevent the overfitting problem. Moreover, prior reports indicate that improper use of dropout in recurrent network models such as LSTM can degrade performance by eroding sequence information [24]; therefore, the dropout rate was set to 0.0.

3.2.3. GRU. GRU is also a type of RNN designed to mitigate the long-term dependency issue of RNNs [25]. The GRU is a simpler version of LSTM, as it is composed of only two types of gates—a reset and an update gate. Thus, the training process is more efficient than the LSTM model. Here, a stacked-GRU architecture was designed similar to the LSTM-based surrogate model. The model consists of two identical GRU layers of 400 units and a hyperbolic tangent activation function.

3.2.4. Combined Network. In addition, combinations of CNN, LSTM, and GRU networks were considered. By connecting the CNN layer and the LSTM or GRU layer, a CNN-LSTM and CNN-GRU network was constructed. Figure 6 illustrates the structure of the proposed CNN-LSTM and CNN-GRU architectures, followed by Table 3 summarizing the network configuration parameters. First, the one-dimensional convolutional layer receives information about the TH variables, component failure, and SAM strategy activation at the previous three time steps to apply a convolution operation. The CNN layer extracts the distinctive features and patterns of the input data and delivers them to the subsequent LSTM/GRU layer. The LSTM/GRU memory cells process the incoming sequential data. Finally, the TH variables at the next time step ($t + 1$) are obtained from the fully connected output layer.

3.3. Training Process. After constructing the DNNs described in the previous section, the model is trained with the training dataset that has been previously split from the MAAP dataset. The details of the training process are explained as follows.

First, the validation scheme should be determined. Two types of validation strategies exist—k-fold cross-validation and hold-out validation. In the k-fold cross-validation scheme, the dataset is divided into “k” groups. Each group is divided into a training set and test set to train and evaluate the models.

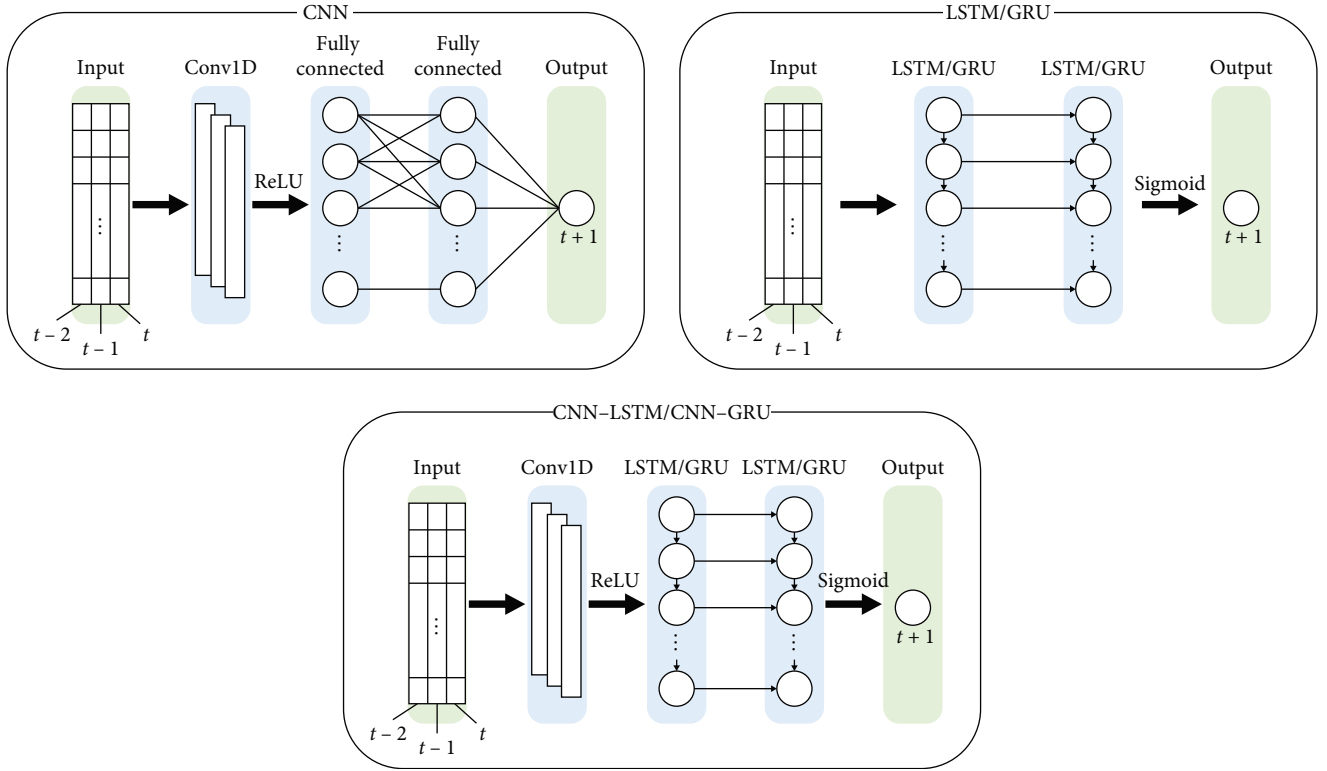


FIGURE 6: Structure of the DNN models.

TABLE 3: Network configuration parameters of the combined networks.

Layer	Hyperparameter
Input	$32 \times 3 \times 20$
Convolution	3×3 size, 400 filters, 1 stride
ReLU	—
LSTM or GRU	400 hidden size, 2 layers, bidirectional = False, dropout = 0.0
Fully connected	10 units
Sigmoid	—

The process is repeated until all “k” groups are used to train the model. This scheme is usually applied to cases suffering from a lack of training data [26].

On the other hand, the holdout validation randomly splits the entire dataset into a training set, validation set, and test set. As the validation set constantly monitors and tunes the hyperparameters, overfitting issues can be prevented, and the model is likely to produce agreeable results on unseen data. This study adopts the hold-out validation method, as the MAAP accident datasets are assumed to be ample enough. It should be noted that there are 10,780 accident scenarios, each composed of time series data for 20 input parameters, whose series length is 73 (from $t = 0$ to $t = 72$ h). Thus, a total of 15,738,800 data points consist of the MAAP accident dataset.

The loss function is the mean squared error (MSE) of the validation set. An Adam optimization algorithm is used, which is a combination of the root mean square propagation and the momentum method. The metrics used to evaluate the model’s performance are mean absolute error (MAE) and RMSE.

To set the end point of the training process, an early stopping module is applied. The module stops the training process when the validation loss has stopped decreasing for more than 50 epochs. Unless the training is stopped by the early stopping module, the training process can be continued for 500 epochs. When the training is stopped, the model restores the best weights from the epoch with the smallest validation loss. All training processes were performed on an Intel Xeon CPU E5-2640 with 64 GB RAM.

3.4. Model Evaluation

3.4.1. Rolling Window Method. To forecast the progression of a 72-h accident scenario, a rolling window method is used. It is one of the time series forecasting methods that calculates a time series on a fixed adjacent window of prior observations and uses it for the forecasting of the next time step. Figure 7 illustrates the schematic of a forecasting process using the rolling window method. Using the past three time steps, the surrogate model predicts the TH variables at the next

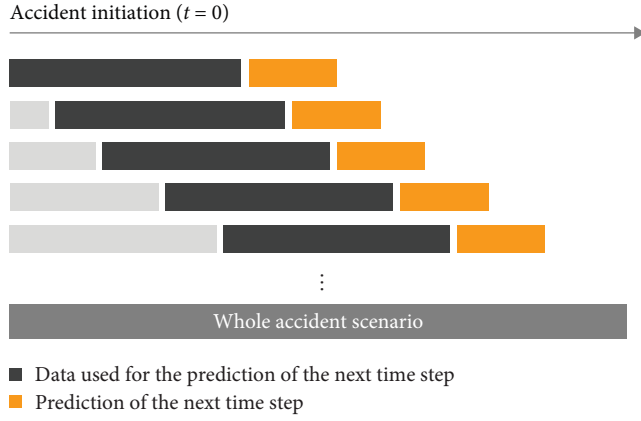


FIGURE 7: Schematic of rolling window method in forecasting the LOCCW accident scenario.

time step. Then, the window is shifted one step forward to the next time step. The surrogate model forecasts the next time step using the updated information at the new window. This process is repeated until the window reaches the end of the scenario (i.e., $t = 72$ h). With this method, it is possible to predict the progression of a whole scenario using only the initial plant states.

In this study, the rolling window method is applied to assess the full-scenario prediction performance of the surrogate models. To check the full-scenario prediction performance of a model, a comparison shall be made between the original MAAP data and the predicted data. The rolling window method is used to produce the full-scenario data using the pre-trained surrogate model.

3.4.2. Error Metrics. The regression performance of the surrogate model was evaluated with MAE and RMSE. Both are popular metrics that represent the error in predicting the plant state at the next time step. MAE helps understand the model's performance over the whole dataset because the value is on the same scale as the predicted target. Meanwhile, the RMSE assigns more penalties to the outliers because the errors are squared during the calculation process. Thus, it is conventional to present both metrics to help the understanding of both the model's performance and the quality of the input data. Equations (1) and (2) describe the calculation of MAE and RMSE, respectively. The difference between the labeled data (y_i) and the predicted data ($\hat{y}_i$) are averaged over " N_{TH} " number of TH variables. To calculate MAE (Equation (1)), the absolute of the differences between y_i and $\hat{y}_i$ are averaged over all TH variables. To calculate RMSE (Equation (2)), the differences between y_i and $\hat{y}_i$ are squared and averaged for all TH variables, and a square root is taken to the averaged value. Thus, the MAE and RMSE describe the general regression performance of a surrogate model to predict the TH variables at the next time step.

$$MAE = \frac{1}{N_{TH}} \sum_{i=1}^{N_{TH}} |y_i - \hat{y}_i|, \quad (1)$$

$$RMSE = \sqrt{\frac{1}{N_{TH}} \sum_{i=1}^{N_{TH}} (y_i - \hat{y}_i)^2}. \quad (2)$$

3.4.3. Dynamic Time Warping (DTW) Distance. To evaluate the full-scenario prediction performance, the DTW distance was calculated. The DTW distance is an intuitive indicator for representing the similarity between two different time series data. While the error metrics (MAE and RMSE) assess how well the surrogate models predict the TH variables at the next time step, the DTW distance evaluates the prediction performance over the whole scenario. If the DTW distance of a TH variable is small, the model is likely to accurately predict the progression of the whole accident progression.

Here, the algorithm for calculating the DTW distance is briefly explained. Suppose that there are two sequential data $P = (p_1, p_2, \dots, p_m)$ and $Q = (q_1, q_2, \dots, q_n)$. The Euclidean distance $d(i, j)$ between the two elements p_i and q_j , is calculated using Equation (3). Then, the warping distances between the sequences are saved in a cost matrix, $D_{m \times n}$. The (i, j) th element of the cost matrix is the sum of $d(i, j)$ and the minimum value among the adjacent elements, $D_{i-1, j}$, $D_{i, j-1}$, and $D_{i-1, j-1}$ (Equation (4)). Repeating this process for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$, all values in the cost matrix are updated.

$$d(i, j) = (p_i - q_j)^2, \quad (3)$$

$$D_{i, j} = d_{i, j} + \min(D_{i-1, j}, D_{i, j-1}, D_{i-1, j-1}), \quad (4)$$

$$DTW \text{ distance} = \sqrt{D_{m, n}}. \quad (5)$$

The goal of the DTW algorithm is to search for a path that minimizes the sum of warping distances. Such a path can be found by backtracking from the last element $D_{m, n}$ to the nearby element that is smaller than the current element. By taking the square root of the last element of the cost matrix, $D_{m, n}$, the DTW distance between P and Q is obtained (Equation (5)).

4. Results and Discussion

In this section, the training results are discussed using the error metrics and DTW distance. The comparisons are made among five different DNN architectures (CNN, LSTM, GRU, CNN-LSTM, CNN-GRU) and three different time resolutions (15, 30, and 60 min).

4.1. Performance Comparison Using Error Metrics. First, the regression performances of the various DNN architectures are compared using the MAE and RMSE values. Table 4 summarizes the MAE and RMSE of various DNN architectures trained with various time-resolution data. It is observed that both MAE and RMSE are relatively small in the combined models (CNN-LSTM and CNN-GRU), followed by CNN, LSTM, and GRU models. Observing the relative differences in MAE and RMSE, the regression performance differences between the CNN-LSTM and CNN-GRU models were

TABLE 4: MAE and RMSE of the five DNN models at various time resolutions.

Model	$\Delta t = 60 \text{ min}$		$\Delta t = 30 \text{ min}$		$\Delta t = 15 \text{ min}$	
	MAE (relative difference, %)	RMSE (relative difference, %)	MAE (relative difference, %)	RMSE (relative difference, %)	MAE (relative difference, %)	RMSE (relative difference, %)
CNN	0.0067 (6.74)	0.0170 (4.25)	0.0058 (6.65)	0.0153 (7.24)	0.0050 (1.80)	0.0140 (7.80)
LSTM	0.0095 (51.56)	0.0349 (113.96)	0.0074 (36.64)	0.0262 (83.71)	0.0053 (7.49)	0.0186 (43.64)
GRU	0.0106 (67.84)	0.0355 (117.87)	0.0081 (49.74)	0.0265 (85.71)	0.0064 (30.57)	0.0197 (51.85)
CNN-LSTM	0.0063 (0.43)	0.0163 (0.12)	0.0060 (11.30)	0.0149 (4.08)	0.0051 (4.08)	0.0129 (0.00)
CNN-GRU	0.0063 (0.00)	0.0163 (0.00)	0.0054 (0.00)	0.0143 (0.00)	0.0049 (0.00)	0.0131 (1.29)

Note: Best performance displayed in boldface. The relative difference in MAE and RMSE compared to the lowest value are displayed together in parentheses.

marginal. The superior performance of the combined models can be explained by the inherent characteristics of each DNN layer. In the combined models, the CNN layer encapsulates the features and patterns inside the input data. The subsequent LSTM/GRU layers process the important features passed on from the CNN layer using its unique gate mechanism. Thus, the CNN layer condenses the number of parameters that must be optimized by the LSTM/GRU layers, while the LSTM/GRU layers allow the addressing of long-term sequential data without falling into the vanishing/exploding gradient problem.

Generally, the RNN-based models (LSTM and GRU) had the largest errors at all time resolutions. However, the difference in the error metrics between the LSTM and GRU models was widened as the time resolution was increased (i.e., the time interval is smaller). The LSTM model tended to outperform the GRU model, and this tendency was strengthened as the time resolution was increased. Such a tendency is expected to stem from the characteristics of the two models' architectures. It has been reported that the LSTM model is excellent at accurately predicting a time series when a massive dataset is given. In contrast, the GRU model is superior at quickly learning the features from a small dataset, since it has fewer gates and parameters than the LSTM model [27]. Applying this to this case, as the time resolution is increased, the size of the dataset is increased, and the LSTM model can be trained more accurately using the expanded dataset.

In addition, it was found that the MAE difference between the LSTM and GRU models was so enlarged that the MAE value of an LSTM model at 15 min was even comparable to that of the combined models. Yet, the RMSE of the LSTM model at 15 min was still large compared to the combined models. The reason why the performance ranking varies depending on the error metrics is because the MAE is usually more robust to the outlier data points. In estimating the MAE, equal weight is given to all errors; thus, the error produced by any data points is reflected in the same proportion. On the other hand, the RMSE takes a square of the error before taking a square root of it. Thus, the RMSE penalizes large gaps between the labeled data and the predicted data. From this background, it can be deduced that the LSTM model can predict the TH variables with higher accuracy on average as the time resolution is increased but cannot successfully minimize the large-scale errors.

Next, the regression performance at various time resolutions is compared. It was observed that as the time resolution increases, the error metrics of the surrogate models decrease. This means that the model's ability to predict the next time step can be enhanced by training the model with high-resolution data. A possible explanation for this trend is that a machine learning model's performance highly depends on the quality and quantity of the training data. As the time resolution is doubled and quadrupled from 60 to 30 min and 15 min, the number of training data is also doubled and quadrupled. Since the models are provided with a larger number of datasets, the trained model's performance is likely to be enhanced.

Interestingly, the performance ranking between the various DNN models was less apparent at higher time resolutions. As the time resolution is increased (the time interval is smaller), the discrepancy between the error metrics of the CNN-based models (CNN, CNN-LSTM, CNN-GRU) and RNN-based models (LSTM, GRU) becomes narrower. The background behind the trend can be explained by the characteristics of the training data and the DNN models. The CNN models are powerful tools for capturing neighborhood information, and RNN models are specialized in handling sequential data. CNNs require adjustments in the size and stride of convolutional filters to effectively learn from higher-resolution data, whereas RNNs inherently adjust to the increased detail through their recurrent structure but need more complex or deeper architectures to handle longer dependencies.

Another explanation for the observed trend may be found in the relationship between the number of training datasets and the trained parameters. The number of parameters that must be optimized at the RNN models is generally larger than CNN models because of the "parameter sharing" property of CNN models [28]. Since the appropriate number of datasets increases with the number of model parameters, it can be inferred that the RNN models typically require a larger number of datasets than the CNN models to achieve a similar performance. Hence, as the number of training datasets increase, the RNN models are fed with a sufficient amount of training data to achieve a similar performance as that of a CNN model. Thus, the differences between the performance metrics of the CNN-based models and the RNN-based models become smaller.

The performance of the surrogate models can also be compared by examining their learning curves. A learning

curve of a machine learning model shows the change in the loss function as the epoch is increased. By comparing the loss of the training set and the validation set, whether the model is overfitted or not can be confirmed. The five figures organized in Figure 8 show the learning curves of the five DNN models at 60-min time resolution. Generally, the difference between the losses of the training set and the validation set was negligible, implying that overfitting is not of concern.

Table 5 compares the lengths of training time of the five DNN models at three different time resolutions. Because the AI model was trained on a CPU rather than on a GPU, the training time was substantially longer. Obviously, the length of training time was increased as the time resolution was increased from 60 to 15 min, as the number of training data was expanded. Notably, the CNN model could terminate the training process in a short time. As the number of parameters that must be optimized is smaller in a CNN layer [29], the lengths of training time of CNN-based models could be shortened. It was also confirmed that by combining a CNN layer to the LSTM model, the training time could be reduced.

Next, the GRU-based models (GRU and CNN-GRU) took slightly shorter time to be trained compared to LSTM-based models (LSTM and CNN-LSTM). The reason for such observation roots from the models' architectures. While the LSTM is composed of three gates (input, output, and forget gates), the GRU is composed of only two gates (reset, update gates). Due to the simplified structure of the GRU, the length of training time can be curtailed. In summary, the CNN-GRU model could be trained in a relatively short time and even showed a superior regression performance, demonstrating that it is a computationally efficient DNN architecture for forecasting an NPP's state during a LOCCW accident.

It should be noted that the most time-consuming step in the study was generating accident data with the MAAP code, whereas the model training itself was comparatively short. Once trained, the surrogate model enables real-time accident prediction. Table 6 presents the average computation time for 1100 test scenarios using the MAAP code and each surrogate model. Each surrogate model forecasts the 72-h scenario using a rolling-window method. Since the models with longer training times contain more parameters, their computation times were longer. Nonetheless, the surrogate models calculate the scenario significantly faster than the MAAP code. Hence, utilizing the surrogate model allows for faster real-time forecasting of severe accidents than direct MAAP calculations, underscoring its value as an accident-management support tool.

4.2. Performance Comparison Using DTW Distance. As explained in Section 3.4.3, the DTW distance between the two time series data $P = (p_1, p_2, \dots, p_m)$ and $Q = (q_1, q_2, \dots, q_n)$ is calculated using a cost matrix of size $m \times n$. Applying this to the cases, the cost matrix sizes for calculating the accident scenario data based on a 15, 30, and 60 min-time resolution are 288×288 , 144×144 , 72×72 , respectively. Thus, the amount of cost accumulated at the last element of the cost matrix for a 30 and 15 min data are 4-times and

16-times bigger than that of a 60 min data, respectively. As the DTW distance is calculated by taking a square root of the last element of the cost matrix, the final DTW distance value of a 30 and 15 min data are 2-times and 4-times larger than that of a 60 min data, respectively. To fairly compare the DTW distances of the three datasets, the DTW distances shall be normalized in this manner.

The number of test data that was utilized in calculating the DTW distance was 1078. The DTW distances are calculated for all 1078 accident scenarios by comparing the labeled data produced by MAAP code and the predicted data produced by each surrogate model. The 1078 DTW distances were averaged to provide a rough estimate of the full scenario prediction performances of each model. The results are plotted in Figure 9, showing the mean DTW distances of 10 TH variables, with their standard deviations expressed in error bars, at each time resolution.

4.2.1. Comparison Between DNN Models. First, the full scenario prediction performances of the different DNN models are compared within a fixed time resolution. If the time resolution is fixed as 60 min, the CNN-LSTM and CNN-GRU models displayed the smallest mean DTW distances at all TH variables except for the CTMT pressure. Especially, the mean DTW distances were significantly smaller for PPS, Max CET, RVWL, and PZR-related variables. The CNN model did not show a clear and general trend in the mean DTW distances. For some TH variables, the CNN model was successful, such as the PPS and the PZR-related variables. However, for other TH variables, the CNN model showed poorer or comparable performances compared to other DNN models, such as the RV WL, SG1 Pressure, Max CET, and CTMT. In addition, the standard deviations of the CNN model were relatively large. Given the inconsistent trend in the mean DTW distances and the large standard deviation values, it can be deduced that the CNN model's full scenario prediction performance is relatively unstable. On the other hand, the RNN-based models (LSTM and GRU) showed the largest mean DTW distances in general, but the standard deviations were relatively small given the large mean values.

However, as the time resolution became higher from 60 to 15 min, there were changes in the trends that have been discussed above. While the combined models generally outperformed the RNN-based models at a 60-min time resolution, the difference in the performances between the combined models and the RNN-based models became smaller as the time resolution was increased to 15 min. For predicting the CTMT pressure, the LSTM model even outperformed the combined models. Similarly, the GRU model outperformed the other models when predicting the RVWL and SGWL. The reason why the full scenario prediction performance ranking between the DNN models has become less apparent can be found in the regression performances of the surrogate models. As mentioned in Section 4.1, the difference in the regression performance between the CNN-based models and the RNN-based models become narrower when the time resolution is increased. As the regression performance for predicting a TH

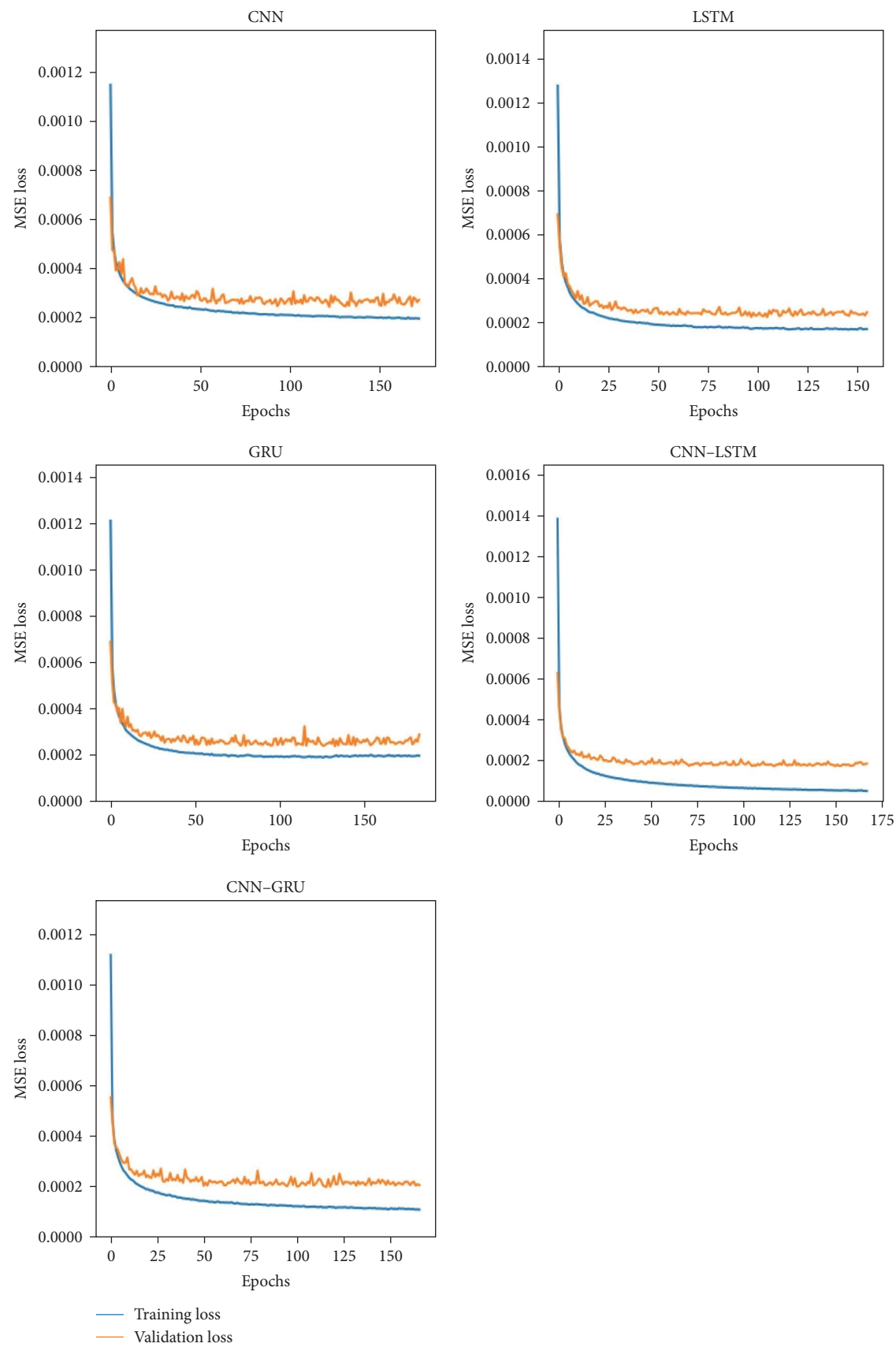


FIGURE 8: Learning curves of the surrogate models.

TABLE 5: Comparison of the training times of the five DNN models at three different time resolutions (Δt).

Model	Training time (h)		
	$\Delta t = 15 \text{ min}$	$\Delta t = 30 \text{ min}$	$\Delta t = 60 \text{ min}$
CNN	0.81	0.44	0.23
LSTM	23.15	7.12	3.99
GRU	9.62	6.93	3.5
CNN-LSTM	12.4	6.36	2.31
CNN-GRU	4.01	3.35	2.03

TABLE 6: Comparison of the average computation times of the MAAP code and the surrogate models for 1100 test scenarios.

Computation tool	Computation time (min)		
	$\Delta t = 15 \text{ min}$	$\Delta t = 30 \text{ min}$	$\Delta t = 60 \text{ min}$
MAAP		75.4	
CNN	0.0097	0.0028	0.0015
LSTM	0.0633	0.0109	0.0061
GRU	0.0321	0.0117	0.0059
CNN-LSTM	0.0782	0.0146	0.0053
CNN-GRU	0.0156	0.0079	0.0048

variable at the next time step affects the full scenario prediction performance, the ranking of DTW distances between various DNN models become less apparent.

4.2.2. Comparison Between TH Variables. Not only does the full scenario prediction accuracy depend on the type of DNN architecture, but it also varies for different TH variables. Generally, the mean DTW distances of PPS, PZR pressure, cold leg temperature, and hot leg temperature were small. It is because the four variables are physically correlated to each other. For example, the PPS is almost the same as PZR pressure because the RCS pressure is controlled by the pressurizer. The cold leg temperature is closely related to the PPS due to the thermo-physical relationship between the pressure and temperature of a fluid. Also, the hot leg temperature is bound to be affected by the pressurizer. Looking at Figure 2, the hot leg temperature is measured at node #16, which is located directly under the pressurizer. As there are multiple input variables that are physically inter-correlated, it is highly likely that the accuracy for predicting those TH variables is enhanced.

On the other hand, the mean DTW distance of RVWL was generally high at all time resolutions. Such inaccuracy may originate from the characteristics of the complex dynamics of the RVWL profile during the LOCCW accident. The RVWL experiences a sharp decrease when the RPV fails. Until then, the water level can remain constant even under multiple component failures. There are also cases where the RVWL is restored to the top core level even after RPV failure. If the HPI or LPI pump is robust, safety injection into the reactor vessel can maintain the RVWL. Also, the activation of SAM-3 strategy (RCS external injection) enables the recovery of RVWL. However, the implementation of a safety injection depends on

the conditions inside the reactor vessel, especially the pressure. When the pressure inside the vessel is too high, for instance, the RV WL cannot be restored even when the LPI pumps are available or the SAM-3 strategy is activated.

To visualize these dynamics, the TH variable is plotted in Figure 10. The two figures show the changes in the TH variables during two different accident scenarios simulated by the MAAP code. The input conditions (component failure times and SAM strategy activation times) are summarized in Table 7. Looking at scenario #150, it is observed that the RV WL is promptly recovered at around 43–45 h, which coincides with the SAM-3 activation time, $t = 43 \text{ h}$. The immediate recovery was possible because the RCS was already sufficiently depressurized at around $t = 43$, referring to the PPS profile. Thus, the RCS injection could be implemented successfully, enabling the rise of RV WL.

On the contrary, scenario #50 is an opposite case where the RCS external injection could not be implemented immediately. Although the SAM-3 strategy was intended to be activated at $t = 19 \text{ h}$, the injection was actually initiated around $t = 21 \text{ h}$. Looking at the PPS plot, the time of a complete RCS depressurization occurred at around $t = 20 \text{ h}$. Hence, there has been a delay between the SAM-3 activation and the rise in RVWL.

From these examples, it can be deduced that the dynamics of RVWL during a severe accident are highly complex, and the intercorrelations between the input variables cannot be simply understood in a uniform manner. While constructing the surrogate models, the switching on/off of the components and SAM strategies has been expressed as 1 or 0. When these binary values are propagated through the DNN models, weights and biases are applied to them. Such simple operations inside the neural networks would be insufficient to imitate the complex dynamics of RVWL that have been discussed above.

4.2.3. Comparison Between Time Resolutions. To evaluate the effect of time resolution on the full scenario prediction performance, the mean DTW distances are compared at different time resolutions. Upon comparing the graphs for $\Delta t = 15$, 30, and 60 min in Figure 9, it is observed that the mean DTW distances do not significantly decrease with an increase in time resolution. Although the ranking between the DNN models was changed, as discussed in Section 4.2.1, the magnitude of the mean DTW distances and the standard deviations did not change notably as the time resolution was increased. This implies that the full scenario prediction performance is not necessarily enhanced as the time resolution becomes higher.

Such a counterintuitive result can be attributed to the build-up of prediction errors caused by repeated computations. The number of required calculations for a given scenario quadruples when the surrogate model changes from a 60-min time resolution to a 15-min one. While enhancing the time resolution has reduced the MAE for each surrogate model, the reduction is insufficient to counterbalance the error that accumulates due to repeated calculations. To support this explanation, the accumulated absolute error between

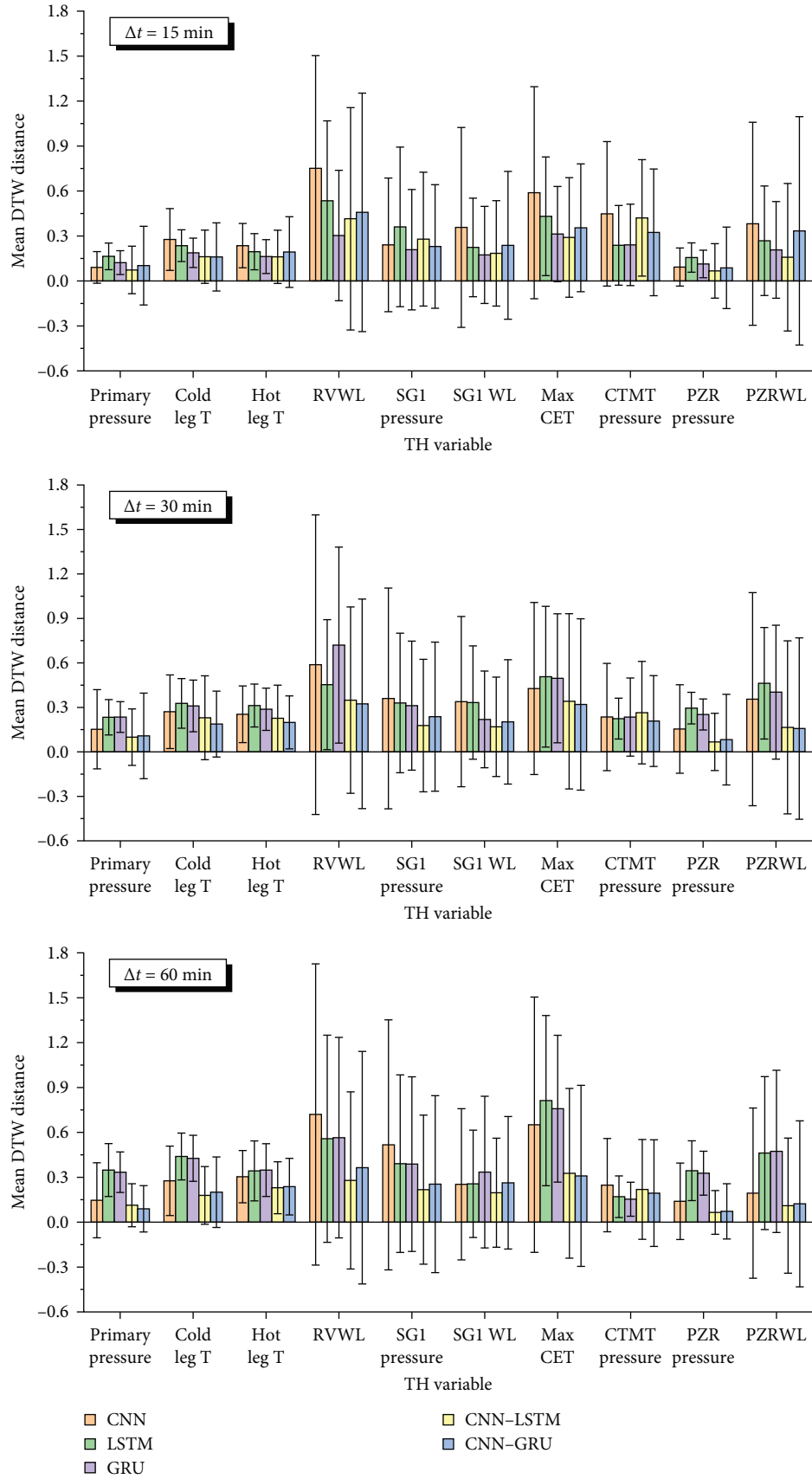


FIGURE 9: Comparison of mean DTW distance of five DNN models at $\Delta t = 15, 30, 60$ min. The error bars represent the standard deviation of the DTW distance distribution.

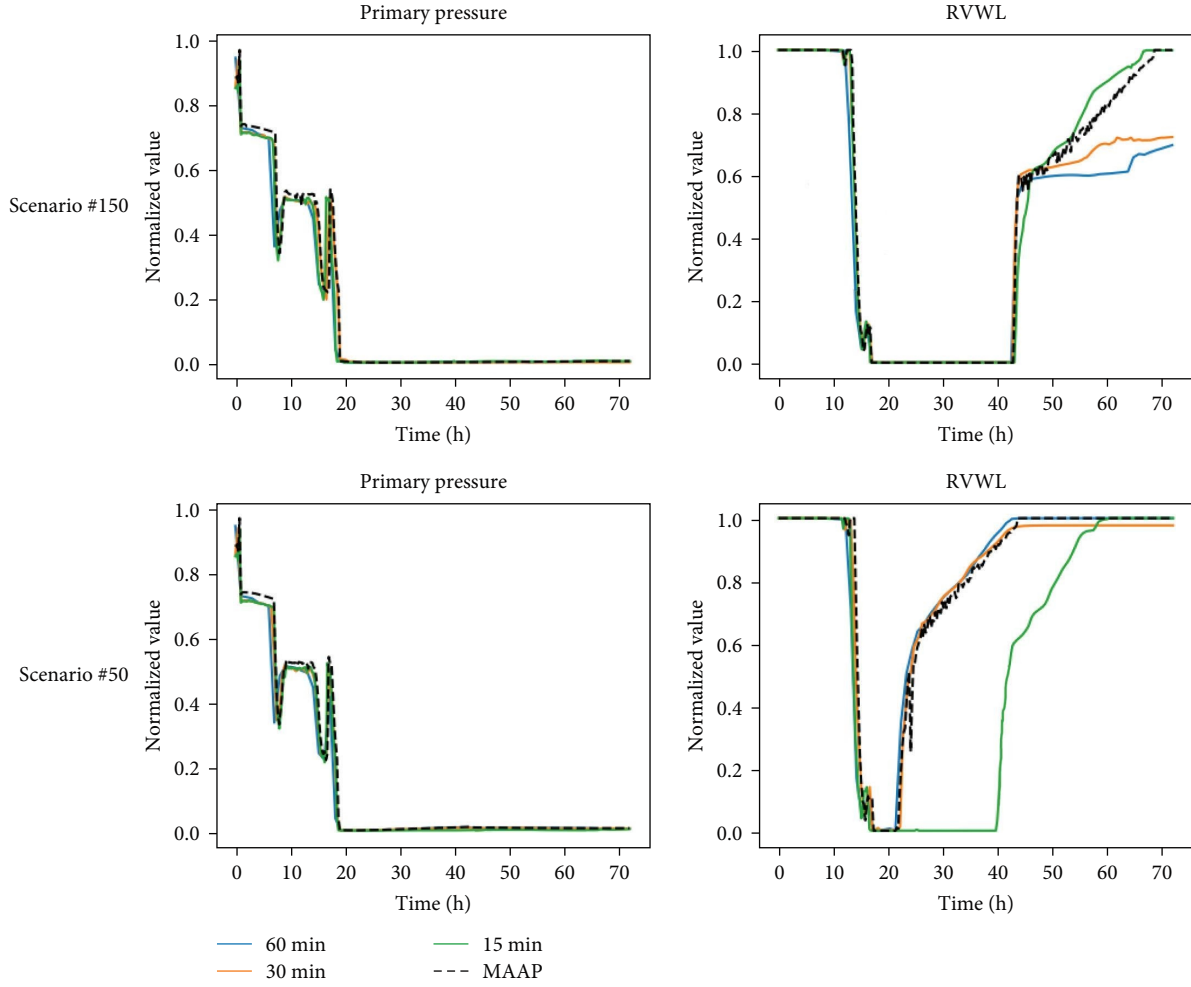


FIGURE 10: Primary pressure and RVWL plots during two different accident scenarios. The graph on the top is the TH variable plot for scenario #150, while the graph on the bottom is for scenario #50.

TABLE 7: Summary of the component failure times and SAM strategy activation times of scenario #150 and #50.

Component and SAM strategy	Scenario #150	Scenario #50
RCP seal LOCA	1	1
HX failure	31	—
HPI pump failure	—	—
LPI pump failure	45	34
CSS pump failure	35	—
MDAFW pump failure	30	—
Charging pump failure	56	2
SAM-1 activation	—	—
SAM-2 activation	64	25
SAM-3 activation	43	19

the predictions made by CNN-LSTM models and the MAAP code is evaluated. Equation (6) shows how the accumulated error of a TH variable i is calculated at a specific time $t = \tau$. It should be noted that the error is calculated on an hourly basis to allow for a fair comparison between different time resolutions. A specific accident scenario was chosen as a test case,

one where the 15-min model generated the lowest mean DTW values for RVWL. This test scenario is denoted as scenario #177.

$$\text{Accumulated error}_i(\tau) = \sum_{t=0}^{\tau} |y_{i, \text{MAAP}}(t) - y_{i, \text{pred}}(t)|. \quad (6)$$

The error accumulation plots in Figure 11. distinctly illustrate that as the accident progresses, these errors accumulate, and their rate of accumulation is notably quicker in the 15-min model. In contrast, in the 60-min model, the accumulated errors tend to rise gradually and consistently over time. Consequently, this confirms the hypothesis that the recurrent calculations have impaired the full scenario prediction performance of high-resolution models. However, these findings do not imply that a lower time resolution invariably degrades predictive performance or that a higher resolution is universally superior. As the large error bars in the DTW-distance plot (Figure 9) demonstrate, performance varies considerably from one scenario to another. In other

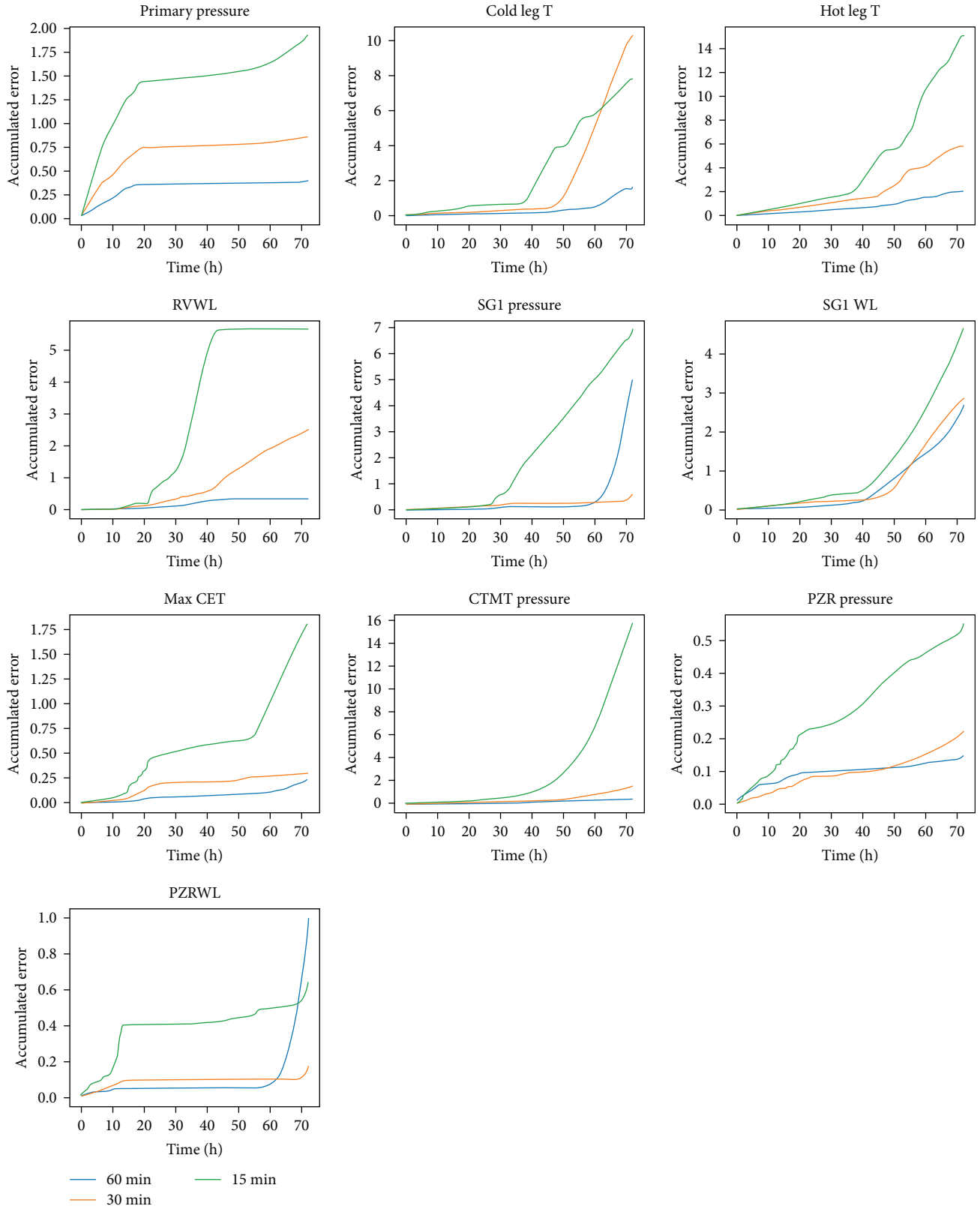


FIGURE 11: Accumulated error of the TH variables for accident scenario #177. For every TH variable, the accumulated error between the MAAP data and the data predicted by 15, 30, and 60 min. CNNCTMTLSTM models are plotted separately for comparison.

words, time resolution is not the sole determinant of prediction performance; rather, scenario-specific dynamics (as exemplified in Figure 10) play a critical role.

In terms of the model's training process, high-resolution data can provide more detailed information about short-term changes, potentially allowing the network to capture more nuanced temporal patterns. However, the risk of overfitting can increase due to larger amount of detail and noise present in the high-resolution data. It requires more complex models to capture the detailed patterns, which in turn may make the model prone to overfitting on the training data.

5. Conclusions

In an accident situation of a NPP, it is a challenging task for the operators to take a prompt mitigation action based on the plant's symptoms. This study delineates the potential of DNNs in forecasting critical parameters during NPP accidents, with a specific emphasis on LOCCW scenarios. By analyzing the predictive accuracy of various DNN architectures across multiple time resolutions, this study offers insights into optimizing DNN applications for severe accident scenarios in NPPs, summarized as follows:

- The CNNCTMTLSTM and CNNCTMTGRU models exhibited the lowest error metrics across all time resolutions. However, as the time resolution was increased, the LSTM model showed a comparable performance to the combined model, implying that the performance of DNNs varies depending on the type of problem and the characteristics of the data. Thus, it is recommended to try various models depending on the characteristics of the application.
- As the time resolution was increased, the MAE and RMSE values decreased at all DNN architectures, possibly due to the increase in the size of the training data. Thus, it was proven that the accuracy for predicting the next time step can be enhanced by increasing the time resolution.
- Among the 10 TH variables, the mean DTW distance of reactor vessel water level (RVWL) was generally high at all time resolutions. The poor prediction ability for RVWL is due to the highly complex plant dynamics during a severe accident.
- Even if the time resolution was increased, the magnitude of mean and standard deviations of DTW distances did not change notably. While enhancing the time resolution has reduced the MAE for each surrogate model, the reduction is insufficient to counterbalance the error that accumulates due to repeated calculations. This leads to the conclusion that using the rolling window method for time series forecasting may harm the benefits coming from increasing the time resolution of the training dataset.

This study is unique in dealing with severe accident scenarios in which various safety component failures and mitigation strategies occur at various time points. Although there

are existing studies that applied DNN techniques for predicting accidents in NPPs, most of the applications are limited to design basis accident scenarios such as LOCA and SGTR. On the other hand, the LOCCW accident considers the failure of multiple components related to core cooling, extending the progression of the accident to reactor core damage. Thus, the range of accident scenarios is wider than the previous studies. Also, this study is practical in that the input parameters to the model are all observable in the MCR. In an accident situation, the operators must make decisions based on the symptoms of the NPP which are displayed at the MCR. The surrogate model proposed in this study is practical, as it can predict the progression of the accident within a few seconds using only the MCR-observed variables.

However, several limitations of this study could be pointed out. The surrogate models were trained based on the data produced from MAAP code, which is a simulation code for analyzing severe accidents. As the training data were not based on a real accident data, the validation of the surrogate model remains challenging. Nevertheless, accelerating the calculation of the MAAP code is meaningful, as the nuclear industry uses the MAAP code to predict the accident. Also, it has been found that prediction accuracy for RVWL needs to be enhanced, since the RVWL is one of the most important parameters related to the reactor core integrity. To achieve this goal, a different scheme for expressing the switching on/off of the components and SAM strategies shall be developed. The development of countermeasures to physically represent the complex intercorrelations and conditions of the plant dynamics will elevate the overall prediction accuracy of the surrogate model. Lastly, as the main conclusion of this study is that the rolling window method can impair the advantages coming from increasing the time resolution of the time series data, alternative methods to forecast the TH variables throughout the accident scenario should be further searched if high time resolution is important for the prediction.

Nomenclature

TLOCCW:	Total-loss-of-component-cooling-water
MAAP:	Modular accident analysis program
RCP:	Reactor coolant pump
RCS:	Reactor coolant system
LOCA:	Loss of coolant accident
AMST:	Accident management support tool
SAMG:	Severe accident management guideline
TH:	Thermal-hydraulic
DNN:	Deep neural network.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

Author Contributions

Semin Joo: conceptualization, methodology, software, formal analysis, investigation, writing – original draft. **Yeonha Lee:** data curation, methodology, software. **Seok Ho Song:** data curation, methodology, software. **Kyusang Song:** supervision, conceptualization, funding acquisition. **Mi Ro Seo:** supervision, conceptualization, funding acquisition. **Sung Joong Kim:** supervision, conceptualization, funding acquisition. **Jeong Ik Lee:** Supervision, conceptualization, writing – review and editing, funding acquisition.

Funding

This work was supported by Korea Hydro & Nuclear Power Co., Ltd. (Grant 2024-Tech-08).

References

- [1] International Nuclear Safety Advisory Group, “Basic Safety Principles for Nuclear Power Plants: 75-INSAG-3 Rev. 1,” (International Atomic Energy Agency, 1999).
- [2] International Atomic Energy Agency, “The Fukushima Daiichi Accident Technical Volume 2/5 Safety Assessment,” (International Atomic Energy Agency (IAEA) (2015).
- [3] International Atomic Energy Agency, “IAEA Report on Severe Accident Management in the Light of the Accident at the Fukushima Daiichi Nuclear Power Plant,” (Vienna, Austria International Experts Meeting, Vienna, Austria).
- [4] M. Saghaei and M. B. Ghofrani, “Accident Management Support Tools in Nuclear Power Plants: A Post-Fukushima Review,” *Progress in Nuclear Energy* 92 (2016): 1–14.
- [5] D. Roverso, “Soft Computing Tools for Transient Classification,” *Information Sciences* 127, no. 3-4 (2000): 137–156.
- [6] F. Antonello, J. Buongiorno, and E. Zio, “Physics Informed Neural Networks for Surrogate Modeling of Accidental Scenarios in Nuclear Power Plants,” *Nuclear Engineering and Technology* 55, no. 9 (2023): 3409–3416.
- [7] H. Gu, G. Liu, J. Li, H. Xie, and H. Wen, “A Framework Based on Deep Learning for Predicting Multiple Safety-Critical Parameter Trends in Nuclear Power Plants,” *Sustainability* 15, no. 7 (2023): 6310.
- [8] S. W. Oh, J. H. Park, H. S. Jo, and M. G. Na, “Development of an AI-Based Remaining Trip Time Prediction System for Nuclear Power Plants,” *Nuclear Engineering and Technology* 56, no. 8 (2024): 3167–3179.
- [9] T. C. H. Nguyen and A. Diab, “Using Machine Learning to Forecast and Assess the Uncertainty in the Response of a Typical PWR Undergoing a Steam Generator Tube Rupture Accident,” *Nuclear Engineering and Technology* 55, no. 9 (2023): 3423–3440.
- [10] N. H. Tiep, J. S. Kim, H. Y. Jeong, et al., “Enhancement of Reflood Test Prediction by Integrating Machine Learning and Data Assimilation Technique,” *International Journal of Energy Research* 2024 (2024): 6446405.
- [11] Y. Lee, S. H. Song, J. Y. Bae, 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* 208 (2024): 110816.
- [12] R. E. Henry, C. Y. Paik, and M. G. Plys, “MAAP4-Modular Accident Analysis Program for LWR Power Plants,” (1994): 3131–3102, Research Project (1994): 3131–3102.
- [13] T.-Y. Kim and S.-B. Cho, “Predicting Residential Energy Consumption Using CNN-LSTM Neural Networks,” *Energy* 182 (2019): 72–81.
- [14] S.-J. Bu and S.-B. Cho, “Time Series Forecasting With Multi-Headed Attention-Based Deep Learning for Residential Energy Consumption,” *Energies* 13, no. 18 (2020): 4722.
- [15] Shin-wolsong 1 & 2, Severe accident management guidelines, Korea Hydro and Nuclear Power.
- [16] Y. P. Zhang, S. P. Niu, L. T. Zhang, S. Z. Qiu, G. H. Su, and W. X. Tian, “A Review on Analysis of LWR Severe Accident,” *Journal of Nuclear Engineering and Radiation Science* 1, no. 4 (2015): 041018.
- [17] S. Joo, Y. Lee, S. H. Song, and J. I. Lee, “Development of an AI-based Loss-of-Component-Cooling-Water Accident Prediction Model,” in *11th Annual Conference of the Korean Institute of Hazardous Materials (KIHM)*, (Korean Institute of Hazardous Materials, 2023).
- [18] S. Joo, Y. Lee, S. H. Song, and J. I. Lee, “Accelerated Prediction of Loss-of-Component-Cooling-Water (LOCCW) Accident Progression,” in *2023 American Nuclear Society (ANS) Winter Meeting and Technology Expo*, (American Nuclear Society, 2023).
- [19] Y. Yamanaka, S. Mizokami, M. Watanabe, and T. Honda, “Update of the First TEPCO MAAP Accident Analysis of Units 1, 2, and 3 at Fukushima Daiichi Nuclear Power Station,” *Nuclear Technology* 186, no. 2 (2014): 263–279.
- [20] J.-S. Yoo and K.-Y. Suh, “Analysis of TMI-2 Benchmark Problem Using MAAP4.03 Code,” *Nuclear Engineering and Technology* 41, no. 7 (2009): 945–952.
- [21] M. I. Radaideh, C. Pigg, T. Kozłowski, Y. Deng, and A. Qu, “Neural-Based Time Series Forecasting of Loss of Coolant Accidents in Nuclear Power Plants,” *Expert Systems with Applications* 160 (2020): 113699.
- [22] Y. LeCun and Y. Bengio, “Convolutional Networks for Images, Speech, and Time Series,” in *The Handbook of Brain Theory and Neural Networks*, 3361, no. 10, (MIT Press, 1995): 3–15.
- [23] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation* 9, no. 8 (1997): 1735–1780.
- [24] Y. Gal and Z. Ghahramani, “A Theoretically Grounded Application of Dropout in Recurrent Neural Networks,” *Advances in neural information processing systems* (2016).
- [25] K. Cho, B. van Merriënboer, C. Gulcehre, et al., “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Association for Computational Linguistics, 2014): 1724–1734.
- [26] S. Yadav and S. Shukla, “Analysis of k-Fold Cross-Validation Over Hold-Out Validation on Colossal Datasets for Quality Classification,” in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, (IEEE, 2016): 78–83.
- [27] S. Yang, X. Yu, and Y. Zhou, “LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset As An Example,” in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, (IEEE, 2020): 98–101.
- [28] D. Dai, L. Yu, and H. Wei, “Parameters Sharing in Residual Neural Networks,” *Neural Processing Letters* 51, no. 2 (2020): 1393–1410.
- [29] G. K. Dziugaite and D. M. Roy, “Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks With Many More Parameters Than Training Data,” *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*.

Copyright © 2025 Semin Joo et al. International Journal of Energy Research published by John Wiley & Sons Ltd. This is an open access article under the terms of the Creative Commons Attribution License (the “License”), which permits use, distribution and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. <https://creativecommons.org/licenses/by/4.0/>