Development of Remaining Trip Time Prediction Algorithm for Abnormal Situations at Nuclear Power Plants

Hyojin Kim[a], Jonghyun Kim[a,*]
[a]*Department of Nuclear Engineering, Chosun University, 10 Chosundae 1-gil, Dong-gu, Gwangju, Republic of Korea, 61452*
[*]*Corresponding author: jonghyun.kim@chosun.ac.kr*

1. Introduction

One of important tasks for operators in abnormal situations at nuclear power plants (NPPs) is to perform urgent actions to prevent the reactor trip and monitor trip-related parameters. However, the decision-making is known difficult due to time pressure and urgency [1]. In addition, sometimes, there is too much information to consider when operators make decisions. NPPs have approximately 4,000 alarms and monitoring devices in the main control room as well as more than one hundred abnormal operating procedures [1]. These information overloads could confuse operators as well as increase the likelihood of errors caused by the increase of mental workload.

Some abnormal operating procedures (AOPs) require operators to perform urgent actions to prevent the reactor trip, even before performing the diagnosis. In addition, operators need to monitor whether the trip parameters reach the trip setpoints, which leads to the emergency operation. Under these situations, adequate situation awareness (SA) affects the effective mitigation of abnormal situations. According to Ensley's three levels of SA, Level 3 SA, which involves predicting future situations, provides the knowledge and time necessary to determine the most favorable course of action to ensure safety [2].

In this light, this study proposes an algorithm to predict the trip parameters and remaining trip time with uncertainty in abnormal situations of NPPs. This algorithm uses transformer encoder, gated recurrent unit (GRU), and conditional variational autoencoder (CVAE). The goal is to predict the long-term trend of trip parameters for 240 time-steps equivalent to 40 min projection with 10 second intervals. The transformer encoder and GRU decoder are used to predict the long-term trend and increase the accuracy of prediction, while CVAE is utilized to estimate the uncertainty of prediction results. This algorithm was trained and implemented using the 3KEYMASTER simulator based on a 1400 MWe pressurized water reactor.

2. Methods

2.1 Transformer Encoder

The transformer was firstly introduced by Vaswani et al., which was mainly based on the attention mechanism [3]. It achieved great success in various natural language processing tasks. The transformer encoder shown in Fig. 1 is composed of stack of N layers with identical structures. Each layer includes two sub-layers: a muti-head attention layer and a fully connected layer. The residual connection and layer normalization are used in each sub-layer to improve the performance. The residual connection allows the model to learn an identified mapping, which helps to preserve important information from the input sequence. Layer normalization helps to improve the stability and convergence of the model by normalizing the inputs to each layer. The combination of the two sub-layers in the transformer encoder allows the model to learn complex representations of the input sequence by attending to different parts of the sequence and mapping them to a higher-dimensional space. The residual connections and normalization help to prevent the vanishing gradient problem and improve the performance of the model [3].

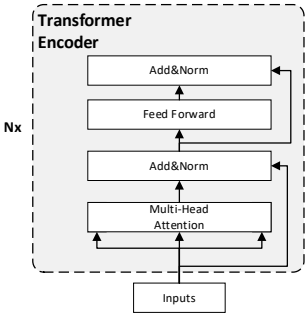

Fig. 1. The architecture of the transformer encoder

2.1.1 Multi-head Attention

The multi-head attention allows the model to learn different aspects of the input representation. It consists of several parallels scaled dot product attention. The scaled dot product attention is the core foundation of the transformer encoder derived from the attention mechanism [3]. It relates different positions of a single sequence to compute a representation. In the scaled dot product attention, an input data $X \in \mathbb{R}^{T \times D}$, where T is the number of input time steps and D is the number of input dimensions, is packed into a query matrix Q_x, a key matrix K_x, and a value matrix V_x. The matrix sizes of Q_x, K_x, and V_x are all $\mathbb{R}^{D \times d_k}$, where d_k is a scaling factor. The scaled dot product attention can be calculated as shown in Eq. (1).

$$Attention(Q_x, K_x, V_x) = softmax\left(\frac{Q_x K_x^T}{\sqrt{d_k}}\right) V_x \qquad (1)$$

The scaled dot product attention is further refined as multi-head attention to jointly attend to the information from different representation subspaces as different positions [3]. In multi-head attention, each attention function is executed in parallel with the respective projected version of the query, key, and value matrix. Then the outputs of all scaled dot product attention functions are concatenated together to produce the final result through a linear layer. The multi-head attention is calculated as shown in Eq. (2).

$$MultiHead(Q_x, K_x, V_x) = Concat(head_1, \dots, head_h)W^O,$$
$$\text{where } head_i = Attention(Q_x W_i^Q, K_x W_i^K, V_x W_i^V) \quad (2)$$

where $i = 1, \dots h$ and W_i^Q, W_i^K, W_i^V, W^O are learnable parameters.

2.2 Gated Recurrent Unit

GRU is a neural network architecture based on the recurrent neural network (RNN) for processing long temporal sequences of data. It simplifies the network structure while ensuring the original accuracy of long short-term memory (LSTM), which has strong ability and efficiency [4]. The architecture of GRU is shown in Fig. 2. Compared with LSTM, GRU mixes the input gate and the forget gate into the update gate and combines the hidden state with the cellular state. The update gate determines how much past information is passed into the future. While the reset gate decides how to combine the new input information with the previous information. The calculation formulas of the GRU are as follows Eqs. (3-6).

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (3)$$
$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (4)$$
$$\widehat{h_t} = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \quad (5)$$
$$h_t = (1 - z_t) r_t \odot h_{t-1} + \widehat{h_t} \odot z_t \quad (6)$$

Here, σ and t represent a sigmoid function and time state, respectively. The cellular state is determined as shown in Eq. (5), where $\widehat{h_t}$ represents the cellular state. Finally, GRU provides the output using Eq. (6), where h_t represents the current hidden state [4].

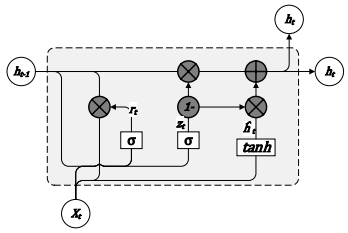

Fig. 2. The architecture of the GRU

2.3 Conditional variational autoencoder

CVAE is an extension of the variational autoencoder to conditional tasks such as translation. Each component of the model is conditioned on some observed x, and models the generation process according to the graphical model. The model distribution is denoted as $p(y|x,z)$, while z is a latent variable with a standard Gaussian prior and factor $p(y|x,z)$. The variational lower bound of the CVAE can be formulated as shown in Eq. (7); where $p_\theta(z|x)$ is the prior model of the CVAE, $q_\phi(z|x,y)$ is the posterior approximator of the CVAE, and $p_\theta(y|z,x)$ is the decoder with guidance from z.

$$L_{CVAE}(\theta, \phi; x, y) = -D_{KL}(q_\phi(z|x,y)||p_\theta(z|x)) + \mathbb{E}_{q_{\phi(z|x,y)}}[logp_\theta(y|z,x)] \quad (7)$$

The first term of Eq. (7) is the Kullback-Leubler divergence between the variational distribution $q_\phi(z|x,y)$ and prior distribution $p_\theta(z|x)$. This term forces the variational distribution to be similar to the prior distribution by working as a regularization term. The second term of Eq. (7) can be understood in terms of the generation of y through the variational distribution $q_\phi(z|x,y)$ and the likelihood $p_\theta(y|z,x)$ [5].

3. Development of a Prediction Algorithm for Trip Parameters and Remaining Trip Time

The overall structure of the prediction algorithm of the trip parameters and remaining trip time is shown in Fig. 3. It comprises 4 functions: 1) pre-processing, 2) long-term prediction, 3) uncertainty estimation, and 4) post-processing. To develop this algorithm, multiple deep learning methods which are transformer encoder, GRU decoder, and CVAE, were applied. The transformer encoder and GRU decoder are used to predict the precise long-term trend and increase the time steps of prediction, while CVAE is applied to assess the uncertainty of prediction results.

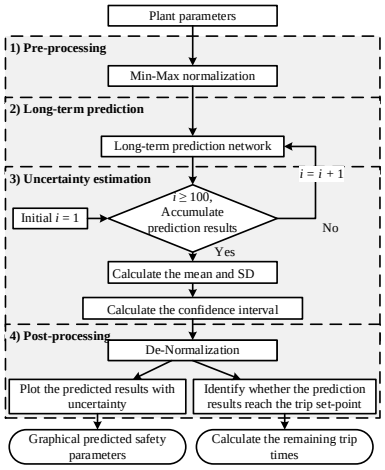

Fig. 3. The architecture of the prediction algorithm.

3.1 Pre-processing Function

The first function of the algorithm is to process the plant parameters and make them suitable as network inputs. The input data for the network should have a range of values from 0 to 1. Generally, variables with higher values will have a larger impact on the network results. However, larger, absolute values are not necessarily more important for prediction than small ones. This problem may produce local minima. Min-max normalization is used to prevent the local minima problem and increase the learning speed. The training data of the network is calculated by using Eq. (8). Here, X is the current value of the plant parameters which are input and output data. X_{min} and X_{max} are the minimum and maximum values of train data, respectively. In this equation, X_{input} has a range of 0 to 1.

$$x_{input} = (x - x_{min})/(x_{max} - x_{min}) \quad (8)$$

3.2 Long-term Prediction Function

This function predicts the long-term trend of variables for trip parameters via a combination of the transformer encoder, GRU decoder, and CVAE. This network is based on an encoder-decoder network to apply for the multi-input and multi-output framework. In addition, a Bayesian model was used to estimate the model uncertainty. This network receives the normalized plant parameters from the pre-processing function and predicts the long-term behaviors of trip parameters. Fig. 4 illustrates the process of the long-term prediction function, which consists of three main components: transformer encoder, variational layer, and GRU decoder.

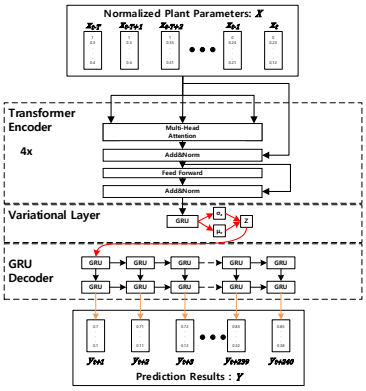

Fig. 4. Process of long-term prediction function.

3.3 Uncertainty Estimation Function

This function is used to estimate the uncertainty of long-term prediction results. It has been implemented with the Bayesian model using conditional variational autoencoder. One of the characteristics of the Bayesian model is that even if the same input data is fed to the network, the network outputs different results. Thus, the prediction uncertainty can be quantified by running several forward passes through the network. This function accumulates the 100 steps of prediction results of the long-term prediction network using the same input data, and the mean and standard deviation (SD) are calculated using Eq. (9) and Eq. (10), respectively, where $Y^{(l)}$ is the prediction results of long-term prediction function at time lth. To set the confidence interval, the upper bound $Upper_Y$ is determined by adding 1.96 σ_Y to μ_Y, and the lower bound $Lower_Y$ is determined by adding -1.96 σ_Y to μ_Y. That is, this function provides uncertainty with a confidence interval that can be 95% certain of the predicted mean.

$$\mu_Y = \frac{1}{100}\sum_{j=1}^{100} Y^{(l)} \quad (9)$$

$$\sigma_Y = \sqrt{\frac{\sum_{l=1}^{100}(Y^{(l)} - \mu_Y)^2}{100}} \quad (10)$$

3.4 Postprocessing Function

The last function is used to make normalized prediction results into suitable units and ranges. Subsequently, the results are also plotted graphically, and then the remaining trip times are evaluated. The first step of this function is to perform the denormalization of the prediction results, i.e., mean prediction values and upper and lower bounds. The denormalization is based on Eq. (11), where $\hat{Y}$ and Y are denormalized and predicted values, respectively. Meanwhile, Y_{min} and Y_{max} are the minimum and maximum values of the output train data, respectively. This function plots the graph using the denormalized mean value and fills the area between the upper and lower bounds for the confidence interval and then presents it.

$$\hat{Y} = Y_{min} + Y * (Y_{max} - Y_{min}) \quad (11)$$

To calculate the remaining trip time, this function checks whether any of the predicted parameters reach the trip setpoint. If the prediction results (i.e., mean prediction values and upper and lower bounds) have reached the trip setpoint, this function identifies which parameters are reached. Then, the remaining trip times are calculated when identified parameters reach the trip set point from the current time.

4. Experiment

4.1 Data Collection

The proposed algorithm was implemented using the 3KEYMASTER simulator based on a 1400 MWe pressurized water reactor. To predict the trip parameters and remaining trip time, we chose the pilot-operated safety relief valve (POSRV) stuck open scenario which can cause the reactor trip. Total of 50 different cases with different opening sizes (1% to 6% at 0.1% intervals) were simulated. All simulations were started from the 100% full-power operation and the data were collected at the one-second interval for one hour. Each scenario injected the malfunction after 10 minutes from starting

the simulation. Among 50 cases of the POSRV stuck open scenarios, 40 cases were used for training while 10 cases were used for testing the algorithm.

4.2 Training

This study selected 10 parameters as the output of the algorithm that are trip parameters of the reference plant. Table I shows the parameters and trip setpoints. To predict 40 minutes of behavior, the time length of the output prediction was selected as 240 steps (i.e., 2,400 seconds). In addition, 40 scenarios (i.e., 12,000 datasets) were used for the training. Among them, 20% of the training data (2,400 datasets) were randomly selected for validation to prevent overfitting.

To train the network, this study used the network input as the 60 input sequences and 715 input parameters. The mean absolute percentage error (MAPE) of validation results was monitored during the training. The MAPE is defined by Eq. (12). Where l and n indicate the index and number of prediction data, respectively. Y^{real} and Y^{pred} indicate the real values and predicted values, respectively. The MAPE of the validation results was 1.23%. In addition. the train loss and validation loss converged into 1.4442E-04 and 9.9872E-05 at 1,326 epochs, respectively.

$$MAPE = \frac{100\%}{n}\sum_{l=1}^{n}\left|\frac{Y_l^{real}-Y_l^{pred}}{Y_l^{real}}\right| \quad (12)$$

Table I: Trip parameters and each trip setpoint.

Trip parameters (Units)	Trip setpoints
Cold-leg #1A, 1B temperature (℃)	Low setpoint: 262.2℃ High setpoint: 310.6℃
Reactor power (%)	High setpoint: 110.4%
Steam generator (SG) #1,2 pressure (kg/cm^2)	Low setpoint: 59.5 kg/cm^2
SG #1,2 narrow level (%)	Low setpoint: 45% High setpoint: 91%
Pressurizer pressure (kg/cm^2)	Low setpoint: 140.7 kg/cm^2 High setpoint: 167.6 kg/cm^2
Containment pressure (cmH_2O)	High setpoint: 124.1 cmH_2O
DNBR	Low setpoint: 1.29

4.3 Result

The test for the trained algorithm was performed by using 10 scenarios (i.e., 3,000 datasets). The MAPE of test results is 1.93%. In general, the values of MAPE below 10% can designate a high accuracy of prediction [2]. The prediction results demonstrated that the proposed algorithm accurately predicted 40 min of trip parameters (i.e., 10 parameters and 240 steps).

Fig. 5 shows the test results of the 40 min prediction of 10 trip parameters with uncertainty estimation and remaining trip times in the case of 5.3% POSRV stuck open. The black and blue lines represent the trip setpoint and the past trends of the trip parameters, respectively. The red and orange lines represent the prediction results by the proposed algorithm and real trends in the test scenarios, respectively. The light gray shaded areas represent the 95% confidence intervals of predicted values.

In addition, since the predicted pressurizer pressure and DNBR reached the trip setpoint, the mean remaining trip times were calculated from the current time as 2,080 seconds and 2,120 seconds, respectively. In addition, the confidence area indicated that the remaining trip times of pressurizer pressure were 2,040 seconds and 2,120 seconds. The interval of DNBR remaining trip time with 95% confidence is from 2,090 to 2,140 seconds from the moment. The real remaining trip time in the scenario was 2,100 seconds. The results demonstrate that the proposed algorithm could accurately predict the trip parameters with 240 steps and the remaining trip time.

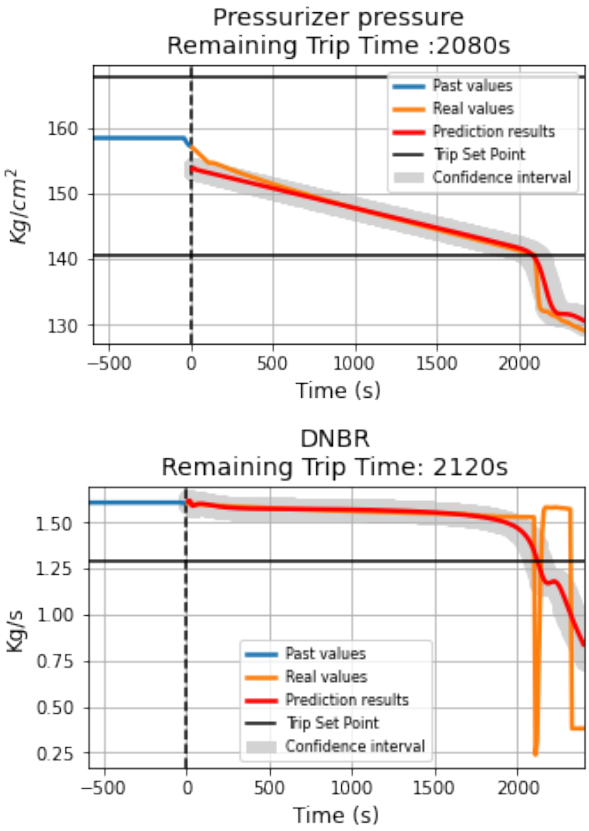

Fig. 5. Predicted results of trip parameters and remaining trip time for 5.3% POSRV stuck open scenario.

4. Conclusions

A novel prediction algorithm that combines the transformer encoder, GRU decoder, and CVAE network was proposed herein to predict long-term trends and provide the remaining trip time. The proposed algorithm not only predicted 10 parameters by 240-time steps equivalent to 40 min but also provided the uncertainty information regarding the prediction results. In addition, the proposed algorithm also calculated the remaining trip

time based on the predicted trip parameters. The validation demonstrated that the proposed algorithm could provide an accurate prediction, as intended. Hence, this algorithm can be applied to an operator support system to improve the SA of operators during abnormal situations in NPPs.

ACKNOWLEDGMENT

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (grant number: RS-2022-00144042).

REFERENCES

[1] J, Kim, S. Lee, P.H. Seong, 'Diagnosis' Autonomous Nuclear Power Plants with Artificial Intelligence, vol 94. Springer Cham, Switzerland (2023)
[2] H. Kim, J. kim "Long-term prediction of safety parameters with uncertainty estimation in emergency situations at nuclear power plants." Nuclear Engineering and Technology. (2023).
[3] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[4] J. Chung, C. Gulcehre, K. Cho, Y. Bengio "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555, (2014).
[5] H. Le, T. Tran, T. Nguyen, S. Venkatesh "Variational memory encoder-decoder." Advances in neural information processing systems, 31, (2018).