Signal Validation Algorithm using Deep Learning Methods

Subong Lee [a], Jonghyun Kim [a*]
[a] *Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, 61452, Republic of Korea*
[*] *Corresponding author: jonghyun.kim@chosun.ac.kr*

1. Introduction

More sensors are used in modern nuclear power plants (NPPs) with the wide application of digital instrument and control (I&C) systems. From the standpoint of safety and reliability, monitoring the condition of sensors is one of the critical tasks because the decision for control action, either by the operator or by the automatic controller, depends on the plant state reflected by sensors.

Traditionally, manual surveillance for sensors and the instrument has been performed but faulty sensors may remain undetected for periods up to the surveillance frequency and unnecessary maintenance can cause a new fault on normal sensors [1]. Hence, on-line monitoring (OLM) techniques of sensors have been an active research area in NPPs. In particular, some researchers have tried to use data-driven models from artificial intelligence (AI) because it can be built on historical operation data without explicitly defined physical mechanisms.

A basic principle using a data-driven model for sensor monitoring is to compare the reconstructed signals from a data-driven model and measured signals. If the reconstructed signal is not similar to measured one, it is considered that the sensor is faulty. To account for this principle, many deep learning-based methods have been developed. One of them is based on the reconstruction error of the generative model. This approach commonly trains a generative model with normal data in the training phase. Then, the reconstruction error generated by the trained model is used to determine normal or abnormal data.

A representative generative model is the Variational AutoEncoder (VAE). The VAE can compute the reconstruction log-likelihood of the inputs modeling the underlying probability distribution of data. The variation generated by VAE is to use existing data to generate potential vector under the encoder, which is subject to Gaussian distribution and can well train the characteristics of the original data so that the generated data will be more reasonable and accurate [2]. However, the original VAE cannot model time series well because time series is usually high dimensional and has the complex temporal correlations. In order to solve this problem, Long Short Term Memory (LSTM) as the encoder and decoder part of the VAE framework can be used to model the time series data [3].

This study suggests an algorithm for the signal validation, based on VAE and LSTM. This algorithm consists of two steps: (1) sensor value reconstruction by VAE-LSTM and (2) detection of fault sensor by comparing the actual sensor value and reconstructed one. The threshold value was also determined to detect the faulty sensor, based on the reconstruction errors. The algorithm was implemented and tested by using a compact nuclear simulator (CNS) developed by Korea Atomic Energy Research Institute (KAERI) is used for acquiring simulation data.

2. Methodology

The proposed algorithm is based on a combined architecture of the VAE and LSTM. This section briefly introduces VAE and LSTM methods.

2.1 VAE

The VAE is an unsupervised deep learning generative model, which can model the distribution of the training data. As shown in Fig. 1, the model's forward propagation process is as follows: the input sample X passes through the encoder to obtain parameters of the latent space distribution. The latent variable z is obtained from sampling in the current distribution, then z is used to generate a reconstructed sample through the decoder [3].

The VAE is a generative model, which is comprised of a probabilistic encoder ($q_\emptyset(z|x)$, recognition model) and a decoder ($p_\theta(x|z)$, generative model). The posterior distribution $p_\theta(z|x)$ is known to be computationally intractable. The VAE approximates $p_\theta(z|x)$ using the encoder $q_\emptyset(z|x)$, which is assumed to be Gaussian and is parameterized by $\emptyset = \{\mu, \sigma\}$, and the encoder learns to predict $\emptyset$. As a result, it becomes possible to draw samples from this distribution.

In order to decode a sample z drawn from $q_\emptyset(z|x)$, to the input x, the reconstruction loss also needs to be minimized. The reconstruction loss is represented by $\mathbb{E}_{q_{\emptyset(Z|X)}}(\log p_\theta(x|z))$. The VAE would approximately map input x into a latent representation deterministically. This gets avoided by minimizing the reconstruction loss together with the Kullback Leibler (KL)-divergence between $q_\emptyset(z|x)$ and prior $p_\theta(z)$. Here, $p_\theta(z)$ is assumed to be multivariate Gaussian $\mathrm{N}(0,1)$. Kingma et al. [4] showed that this loss function is a variational lower bound on log-likelihood of x, i.e., $\log p_\theta(x)$ [5].

$$\mathrm{L}(\emptyset, \theta, \mathrm{x}) = -\mathbb{E}_{q_{\emptyset(Z|X)}}(\log p_\theta(x|z)) + \mathrm{KL}(q_\emptyset(z|x) \parallel p_\theta(z)) \leq \log p_\theta(x) \tag{1}$$

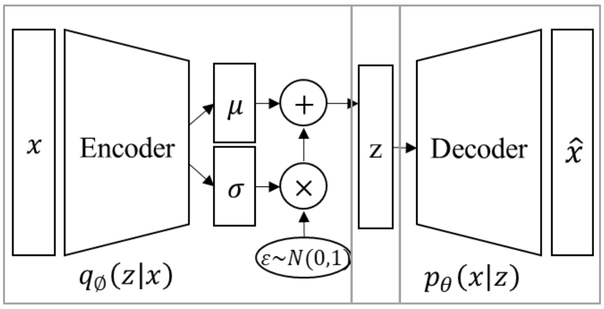

Fig. 1. Architecture of the VAE

2.2 LSTM

The most distinctive feature of LSTM is the gate structure, which appears in the LSTM cell architecture as shown in Fig. 2. The cell state is an essential part of the LSTM. It passes through the whole like a conveyor belt, and the information can continue to pass to the next level without change. Gates are used to update or exclude information based on this cell state. Through the input modulation (g_t) and the input gate (i_t), the LSTM regulates the degree to which the input is updated to the cell state. Eq. (2) represents the input conditioning node and has $\tanh(\emptyset)$ activation function. Eq. (3) represents the input gate and has sigmoid (σ) activation function. Through the sigmoid activation function, it outputs a value of 0 or 1, which determines whether each component will be affected. The forget gate (f_t) and output gate (o_t) are represented by Eq. (4) and (5). This gating structure allows the cell state to control the influence of previous state information on the current state, update the information associated with the current input, and determine the influence level on the output through gate modulation [6].

$$g_t = \emptyset(W_g \cdot [h_{t-1}, x_t] + b_g) \quad (2)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

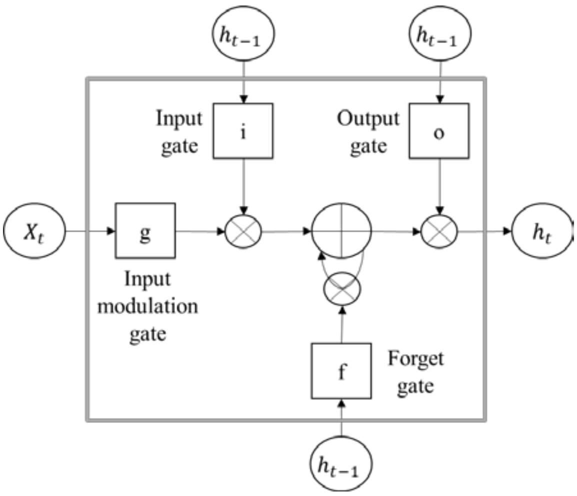

Fig. 2. Architecture of the LSTM cell

3. Algorithm for Signal Validation

3.1 Algorithm modeling

The overall suggested process of the signal validation by using VAE-LSTM is shown in Fig. 3. In this study, a total of 10 sensors are considered, i.e., RCS loop #1, #2, and #3 average temperature, pressurizer pressure, feed water line #1, #2, #3 flow, loop #1, #2, and #3 steam generator level.

Input preprocessing is performed to convert the sensor values to get better performance. Because the sensor values have the different scales, normalizing the value can help prevent convergence at the local minimum. The min-max normalization method [7] is applied to scale the input values for the input layer of the VAE-LSTM model. The minimum and maximum are determined within the input values. Input values are calibrated within the range of 0 to 1 through Eq. 6.

$$Xnorm = \frac{(X - X_{min})}{(X_{max} - X_{min})} \quad (6)$$

The structure of VAE-LSTM proposed in this study is shown in Fig. 3. The selected sensor's data are input into the encoder. The encoder consists of 5 layers with an input layer (10 time steps and 10 nodes), LSTM layer (60 nodes), σ layer (30 nodes), μ layer (30 nodes), and z layer (30 nodes). The input layer is connected to an LSTM layer so that input values pass through an LSTM layer. An LSTM layer outputs means and variances. Then, the latent variables z are obtained from sampling in z layer. The decoder consists of two LSTM layers (10 time steps and 60 nodes, and 10 time steps and 10 nodes). Finally, the latent variables z are decoded by using two LSTM layers. The loss function is Eq. 1. For optimizer, Adam [8] is used, which is the most commonly used optimizer at present. After the training, the proposed VAE-LSTM outputs the reconstructed 10 sensors data.

The output post-processing calculates the residual which is the square of the error between the input and the reconstructed output obtained from the well-trained model. A large residual can indicate that the senor may be faulty. For instance, for the train dataset that include only normal sensor data, a trained model will produce a low residual since it was already trained to reconstruct the data very well. On the contrary, for the input from a faulty sensor, the trained model will produce high residual since it is not trained to reconstruct faulty sensor data. Therefore, it is necessary to define a threshold to determine normal or faulty sensor.

To define the threshold, this study applied a method suggested by W.A. Shewhart [9], based on the reconstruction errors. The basic characteristics of a Shewhart chart are the Center Line (CL), the Upper Control Limit (UCL), and the Lower Control Limit (LCL) in Eq. 7, 8, and 9. The μ and σ mean the mean value and standard deviation of each residual.

$$\mathrm{UCL} = \mu + 3\sigma \quad (7)$$
$$\mathrm{CL} = \mu \quad (8)$$
$$\mathrm{LCL} = \mu - 3\sigma \quad (9)$$

Faulty sensor is detected by comparing the residual with a predefined threshold. If the residual exceeds the predefined threshold, it means that the sensor is faulty.

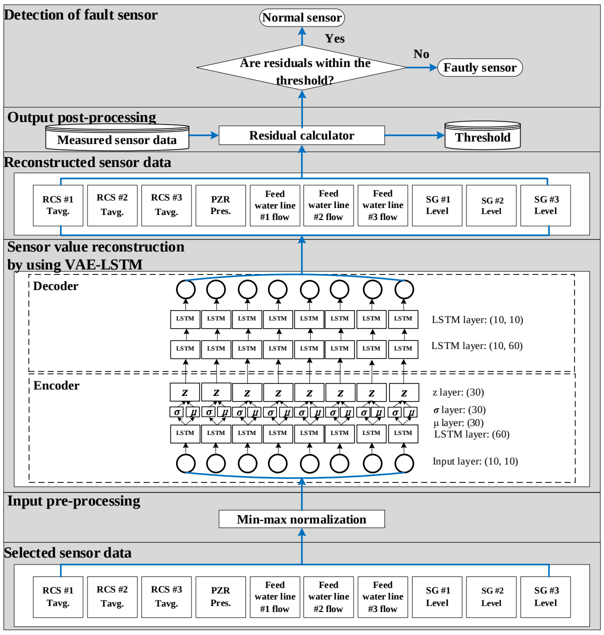

Fig. 3. Algorithm for signal validation

4. Experiments

4.1 Dataset preparation

Datasets are prepared from the 10 sensor variables that are acquired during full power steady-state from CNS. Datasets are divided into train, validation, and test dataset. The train and validation dataset only includes normal sensor data. For the test dataset, it consists of normal sensor data and faulty sensor data.

4.2 Training

The VAE-LSTM is trained using the train dataset. The train dataset with 4990 samples and validation dataset with 2000 samples including normal sensor data are used. Based on the train dataset, the VAE-LSTM model is trained to reconstruct inputs as outputs. The training aim is to minimize the loss function discussed in Section 2.2. The model hyper parameters are set by random grid search method because there is no golden rule for hyper parameter determination to optimize the model [10]. As the hyper parameters, length of latent, batch size, and epochs are 30, 32, and 1000. As shown in Fig. 4, the training result shows that the loss value on the train and validation dataset seems to converge nicely.

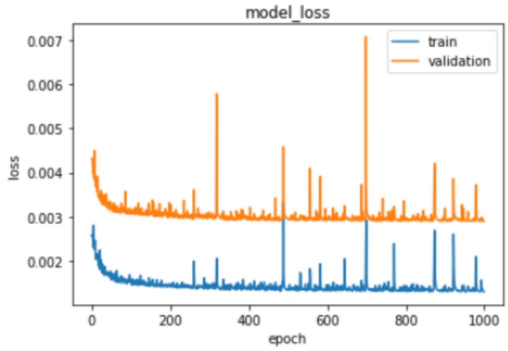

Fig. 4. Training result of the VAE-LSTM

4.3 Test result

Feed water line #1 flow sensor is used as a test signal to validate the fault detection performance. For simulated fault, feed water line #1 flow decreased to 0.02%/sec from 510 to 800 s. The normal signal, the faulted signal, and the estimated signal using VAE-LSTM are shown in Fig. 5, which compares the normal signal and the estimated signal to show that the normal signal is well reconstructed by the sensor value reconstruction function using VAE-LSTM. The second plot in Fig. 5 shows residual values generated by differencing the faulted and estimated signal. The threshold of feed water line #1 flow sensor is defined by Shewhart control limits described in Eq. (7), (8), and (9) (i.e., UCL = 0.0148 and LCL = -0.0144). By predefined threshold, faulty feed water line #1 flow sensor is detected from 510 to 800 s aside from peak points at 430 s and 1050 s.

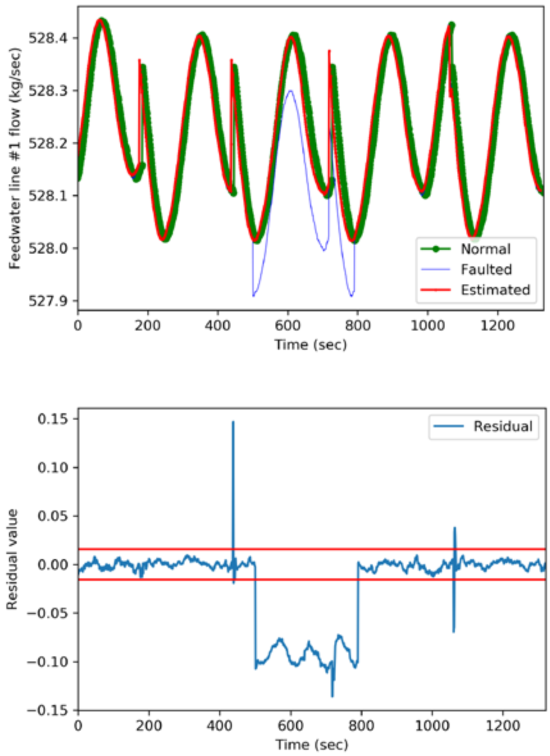

Fig. 5. Test result for the faulty sensor detection

4. Conclusions

This study suggests an algorithm for the signal validation, based on VAE and LSTM. This algorithm consists of two steps: (1) sensor value reconstruction by VAE-LSTM and (2) detection of fault sensor by comparing the actual sensor value and reconstructed one. The threshold value was also determined to detect the faulty sensor, based on the reconstruction errors. The algorithm was implemented and tested by using a compact nuclear simulator (CNS) developed by Korea Atomic Energy Research Institute (KAERI) is used for acquiring simulation data. The test result shown that VAE-LSTM based signal validation algorithm is able to reconstruct the normal sensor and detect the faulty sensor. However, the optimization of the model should be continued so that VAE-LSMT can reconstruct the sensor's behavior precisely. In addition, it is necessary to define more detailed criteria of the threshold because the performance of faulty sensor detection depends on the threshold.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (N01190021-06).

REFERENCES

[1] J. W. Hines and R. Seibert, Technical Review of On-Line Monitoring Techniques for Performance Assessment, Volume 1, 2006.
[2] D. Park, Y. Hoshi, C. C. Kemp, A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-based Variational Autoencoder, 2018.
[3] C. Zhang, S. Li, h. Zhang, and Y. Chen, VELC: A New Variational AutoEncoder Based Model for Time Series Anomaly Detection, 2019.
[4] D. P. Kingma and M. Welling, Auto-encoding variational bayes, International Conference on Learning Representations (ICLR), 2014.
[5] M. Patidar, P. Agarwal, L. Vig, and G. Shroff, Correcting Linguistic Training Bias in an FAQ-bot using LSTM-VAE, 2017.
[6] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Computation, p.1735–1780, 1997.
[7] Y. K. Jain and S. K. Bhandar, Min max normalization based data perturbation method for privacy protection, International Journal of Computer & Communication Technology, 2011.
[8] D. P. Kingma, J. L. Ba, Adam: A Method for Stochastic Optimization, 2014.
[9] W.A. Shewhart, Economic Control of Quality of Manufactured Product, New York: Van Nostrand.[.ISBN 0-87389-076-0], 1931.
[10] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13(Feb), p. 281–305, 2012.