Nuclear Engineering and Technology 53 (2021) 2547–2555

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Leak flow prediction during loss of coolant accidents using deep fuzzy neural networks

Ji Hun Park [a], Ye Ji An [a], Kwae Hwan Yoo [b], Man Gyun Na [a, *]

[a] Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, Republic of Korea
[b] Korea Atomic Energy Research Institute, 989-111 Daedeok-daero Yuseong-gu, Daejeon 34039, Republic of Korea

ARTICLE INFO

Article history:
Received 17 June 2020
Received in revised form
14 November 2020
Accepted 31 January 2021
Available online 8 February 2021

Keywords:
Artificial intelligence
Deep fuzzy neural network (DFNN)
Genetic algorithm (GA)
Loss of coolant accident (LOCA)
Leak flow prediction

ABSTRACT

The frequency of reactor coolant leakage is expected to increase over the lifetime of a nuclear power plant owing to degradation mechanisms, such as flow-acceleration corrosion and stress corrosion cracking. When loss of coolant accidents (LOCAs) occur, several parameters change rapidly depending on the size and location of the cracks. In this study, leak flow during LOCAs is predicted using a deep fuzzy neural network (DFNN) model. The DFNN model is based on fuzzy neural network (FNN) modules and has a structure where the FNN modules are sequentially connected. Because the DFNN model is based on the FNN modules, the performance factors are the number of FNN modules and the parameters of the FNN module. These parameters are determined by a least-squares method combined with a genetic algorithm; the number of FNN modules is determined automatically by cross checking a fitness function using the verification dataset output to prevent an overfitting problem. To acquire the data of LOCAs, an optimized power reactor-1000 was simulated using a modular accident analysis program code. The predicted results of the DFNN model are found to be superior to those predicted in previous works. The leak flow prediction results obtained in this study will be useful to check the core integrity in nuclear power plant during LOCAs. This information is also expected to reduce the workload of the operators.

1. Introduction

According to the operational performance information system for nuclear power plants (NPPs) [1], 123 cases of primary system failure occurred from 2002 to 2019. Among them, 22 cases were events of primary system coolant leakage, including 20 mechanical defects, one human error, and one pipe integrity threat from thermal insulation fire [1]. The pipe integrity gradually degrades over time owing to degradation mechanisms, such as flow-acceleration corrosion and stress corrosion cracking. Accordingly, the frequency of leakage is expected to increase over the lifetime of an NPP.

In the event of an accident in an NPP, proper and prompt action by the operators is crucial to prevent severe accidents. However, to take appropriate actions, the operators must diagnose and respond to several variables early. Furthermore, there are additional considerations to account for when an accident becomes severe. Severe accidents have been rare in NPPs. If an actual severe accident occurs, the operator may have additional workloads, which can lead to human errors. In addition, in case of a severe accident, the variables change rapidly, and the uncertainty of the instrument increases with time. During severe accidents, these changes intensify human errors. This study was conducted to reduce the human error using the artificial intelligence methodology to provide the operator with prediction information.

To acquire the data required by the artificial intelligence algorithm, in this study, an optimized power reactor-1000 (OPR-1000) was simulated using a modular accident analysis program (MAAP) code. The deep fuzzy neural network (DFNN) model was used to predict the leak flow, and the used data were the time elapsed since the start of the event and the leak flow acquired from MAAP code simulations. Furthermore, because the break size is expected to be strongly related to the leak flow, it was used as an input variable to the DFNN model. The break size cannot be measured directly, but it can be estimated accurately [2]. Therefore, the estimated break size was used. The input values were scaled down to consider the uncertainty of the sensor in severe circumstances and to improve the DFNN performance.

* Corresponding author.
E-mail address: magyna@chosun.ac.kr (M.G. Na).

https://doi.org/10.1016/j.net.2021.01.040

2. DFNN model

Cascaded fuzzy neural networks (FNNs) [2] and simplified cascaded FNNs [3] are defined as DFNNs. However, the DFNN model used in this study is a more efficient simplified cascaded FNN-type model. The name of the DFNN model originates from a combination of deep neural networks and FNNs [4]. Briefly, it has a deeply stacked structure of the FNN module. In the rest of this paper, the FNN module, which is the core of the structure, is described first; next, the difference between the DFNN model and the deep neural network (DNN) is explained, and finally the overall structure of the DFNN model is described.

2.1. FNN module

FNN combines the fuzzy inference system (FIS) and neural networks. The FISs are divided into Mamdani-type and Takagi–Sugeno-type fuzzy models. Because the Mamdani fuzzy model generally uses the centroid technique for the defuzzification of the fuzzy output variable, the computation is complicated; however, it is possible to derive more intuitive values. On the contrary, because the Takagi–Sugeno fuzzy model generates the real value directly and does not require defuzzification, the computation is simple and fast [5,6]. In this study, the Takagi–Sugeno fuzzy model was applied on representative real non-linear data from NPPs. The fuzzy rule is expressed as follows:

$$\begin{aligned} &\textit{If } x_1(k) \textit{ is } \phi_{i1}(k),\ \cdots,\ x_m(k) \textit{ is } \phi_{im}(k),\\ &\textit{then } \hat{y}_i(k) \textit{ is } f_i(x_1(k),\ \cdots,\ x_m(k)), \end{aligned} \tag{1}$$

where $x_1,\ \cdots,\ x_m$ are the FNN module input variables, m is the number of input variables, $\phi_{i1},\ \cdots,\ \phi_{im}$ are the fuzzy sets of the i-th fuzzy rule, and $\hat{y}_i$ is the output of the i-th fuzzy rule. The function $f_i(x_1(k),\ \cdots,\ x_m(k))$ in Eq. (1) is expressed as follows:

$$f_i(x_1(k),\ \cdots,\ x_m(k)) = \sum_{j=1}^{m} p_{ij}x_j(k) + r_i\ , \tag{2}$$

where p_{ij} and r_i are the weight and bias, respectively.

The FNN serves as a module in the DFNN model. The FNN module consists of six layers, as shown in Fig. 1. The first layer consists of the nodes that receive the input values and send them to the second layer using the membership function. In the second layer, the membership values are calculated using Eq. (3) and transferred to the third layer. In addition, the membership function uses a symmetric Gaussian function, which is expressed as follows:

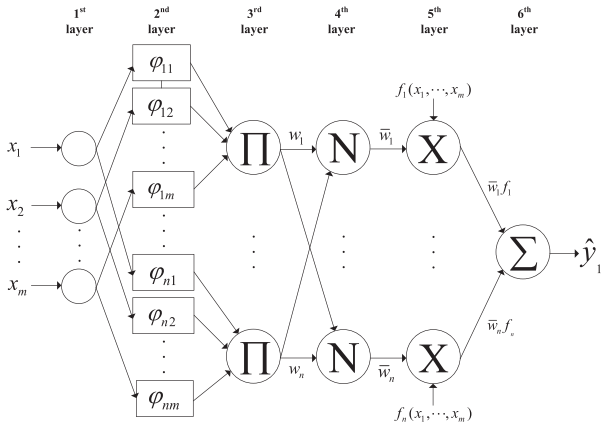


Fig. 1. First FNN module.

$$\phi_{ij}(x_j(k)) = e^{-\left(x_j(k)-c_{ij}\right)^2/2s_{ij}^2}, \tag{3}$$

where c_{ij} is the central position of the symmetric Gaussian function, and s_{ij} is the value that controls the width of the symmetric Gaussian function shape. In the third layer, Eq. (4) is used to multiply all the membership function values and send them to the fourth layer:

$$w_i(k) = \prod_{j=1}^{m} \phi_{ij}(x_j(k)). \tag{4}$$

In the fourth layer, normalization is performed using Eq. (5), and the normalized values are sent to the fifth layer:

$$\overline{w}_i(k) = \frac{w_i(k)}{\sum_{i=1}^{n} w_i(k)}. \tag{5}$$

In the fifth layer, each normalized weight is multiplied by the outputs of the fuzzy rules and sent to the sixth layer. The sixth layer represents the set of all fuzzy if–then rules and the output value is calculated as follows:

$$\hat{y}(k) = \sum_{i=1}^{n} \overline{w}_i(k)y_i(k) = \sum_{i=1}^{n} \overline{w}_i(k)f_i(x_1, x_2, \cdots, x_m). \tag{6}$$

Finally, the output values $\hat{y}(k)$ in the first FNN module are expressed as follows:

$$\hat{y}(k) = \mathbf{w}^T(k)\mathbf{p}, \tag{7}$$

where

$$\mathbf{w}(k) = \Big[\overline{w}_1(k)x_1(k)\ \cdots\ \overline{w}_n(k)x_1(k)\ \cdots\ \overline{w}_1(k)x_m(k)\ \cdots\ \overline{w}_n(k)x_m(k)\ \overline{w}_1(k)\ \cdots\ \overline{w}_n(k)\Big]^T,$$

$$\mathbf{p} = [p_{11}\cdots p_{n1}\cdots p_{1m}\cdots p_{nm}\ r_1\cdots r_n]^T.$$

Currently, vector $\mathbf{p}$ is a conclusion parameter vector, and the vector $\mathbf{w}(k)$ is composed of the values of the input variables and the

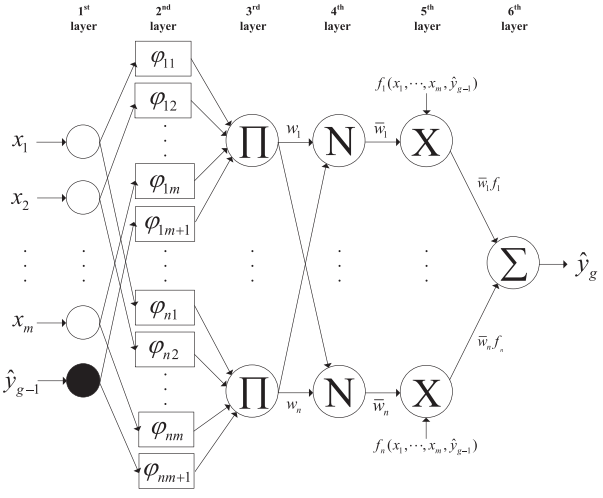


Fig. 2. g^{th} FNN module.

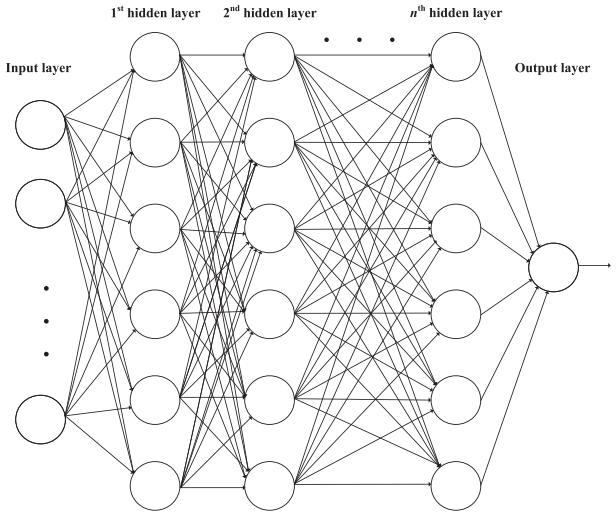


Fig. 3. General structure of the deep neural network (DNN) model.

normalized values of the membership functions. For N_t input and output training data pairs, both $\mathbf{p}$ and $\mathbf{w}(k)$ have $(m+1)n$ dimensions.

Finally, the output vector of N_t input and output data pairs are expressed as follows:

$$\hat{\mathbf{y}}_t = \mathbf{W}_t\mathbf{p}, \tag{8}$$

where $\hat{\mathbf{y}}_t = [\hat{y}(1) \ \cdots \ \hat{y}(N_t)]^T$ and $\mathbf{W}_t = [\mathbf{w}(1) \cdots \ \mathbf{w}(N_t)]^T$.

The DFNN model sequentially adds the FNN module to enhance the prediction performance.

2.2. DFNN model

The DFNN model consists of a contiguous batch of FNN modules. When the input variable values enter the first FNN module, it is optimized through a series of calculations. The output values from the first FNN module are passed together with the original variable values as input values of the second FNN module, where further calculations are performed. By doing this up to the last FNN module, more precise prediction performance and optimization are achieved. This process is illustrated in Fig. 2. Figs. 2 and 4 show that the structure of the DFNN model is similar to that of the DNN model; the difference between the DFNN and the DNN is that their performances are determined by different factors. The performance

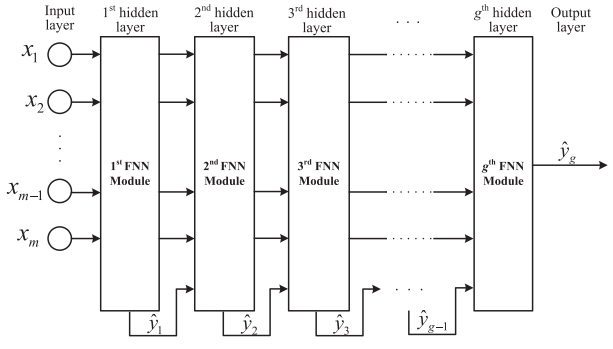


Fig. 4. Structure of the DFNN model.

of the DFNN model depends on the fuzzy rule number and hyper-parameters involved in the membership functions in the FNN module, whereas the performance of the DNN model depends on the number of layers and nodes and on the activation functions (see Fig. 3).

The Takagi–Sugeno type FIS used in a single FNN module is expressed as in Eq. (1). However, the FIS for the g^{th} serial module of the DFNN model used in this paper is expressed as follows:

$$\begin{aligned} &\textit{If } x_1(k) \text{ is } \phi_{i1}^g(k), \ \cdots, \ x_m(k) \textit{ is } \phi_{im}^g(k) \\ &\textit{AND } \ \hat{y}_{(g-1)}(k) \ \textit{ is } \phi_{i(m+1)}^g(k), \\ &\textit{then } \ \hat{y}_g(k) \ \textit{ is } f_g(x_1(k), \cdots, x_m(k), \hat{y}_{(g-1)}(k)), \end{aligned} \tag{9}$$

where $\hat{y}_i$ is the output of the i-th fuzzy rule, and g is the serial module number of the DFNN.

In the single FNN module, the output values are calculated in the FIS. The performance of the DFNN model is improved over that of the FNN model with a single FNN module by passing the output of the very pre-stage module to the input of the next module, as shown in Eq. (9). By comparing Figs. 1 and 2, as the pre-stage FNN module output is added as the input value, the membership

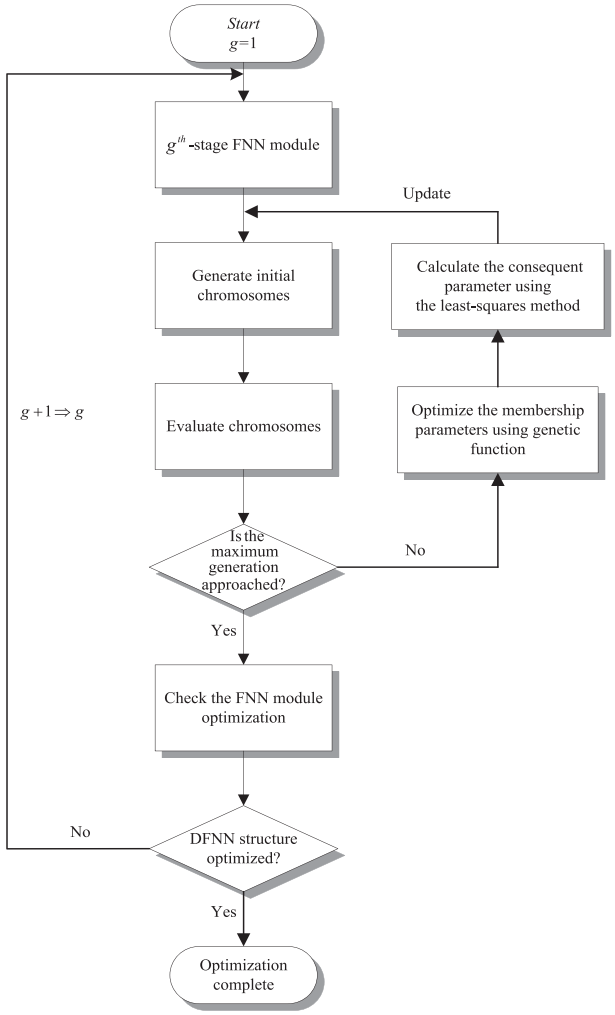


Fig. 5. Optimization procedure for the DFNN model.

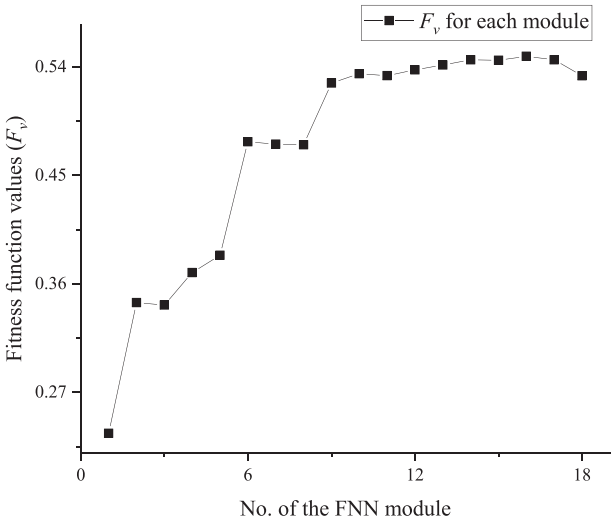


Fig. 6. Fitness function values (F_v) for the verification of data according to the FNN module number.

function is also added and calculated. Fig. 4 illustrates the DFNN model.

2.3. Training the DFNN model

The DFNN model is established by training the model through development data [7] by applying a genetic algorithm (GA) and a least-squares method. The GA is used to improve the performance of the DFNN model by optimizing critical parameters, e.g., c_{ij} and s_{ij}, which determine the shape of all the membership functions. These parameters are encoded and constructed as chromosomes. In this study, the GA parameters were set as follows: population size of 20, mutation probability of 20%, and crossover probability of 100%. The DFNN model was optimized sequentially until the last module in the FNN modules, and the optimized values were updated continuously. The GA minimizes the maximum error and root mean square (RMS) error. For this purpose, the fitness function shown in Eq. (10) was used for measuring the optimization degree. In Eq. (10), relative errors relative to the maximum leak flow at the corresponding break size were used because the usual relative error may not be calculated when the leak flow value is zero or close to zero.

$$F_t = \exp(-\upsilon_{t1}E_{t1} - \upsilon_{t2}E_{t2}), \tag{10}$$

where

$$E_{t1} = \sqrt{\frac{1}{N_t}\sum_{k=1}^{N_t}\left(\frac{y(k)-\hat{y}(k)}{y_{\max}(k)}\right)^2},$$

$$E_{t2} = \max_k\left|\frac{(y(k)-\hat{y}(k))}{y_{\max}(k)}\right|,\ k = 1,\cdots,N_t,$$

υ_1 and υ_2 are the weighting values for the RMS and maximum errors, respectively, and N_t is the number of training data.

After the parameters of the membership functions are determined through the GA, the conclusion parameters are combined and expressed as in Eq. (8). Conclusion parameter **p** is optimized by the least-squares method, which minimizes an objective function defined by the squared errors between the target values and the predicted values [8]:

Table 1
Input signals used to predict the LOCA break size in Ref. [2].

Break position	Input signals
Hot-leg Cold-leg	Pressure and temperature in the containment Pressure and water level in the pressurizer Pressure in a broken side S/G
SGT	Temperature in the containment Water level in the RPV Pressure and water level in a broken side S/G Water temperature in an unbroken side S/G

RPV: reactor pressure vessel, S/G: steam generator, SGT: steam generator tube.

$$S = \sum_{k=1}^{N_t}\left(y(k)-\hat{y}(k)\right)^2 = \sum_{k=1}^{N_t}\left(\mathrm{y(k)} - \mathbf{w}^{\mathrm{T}}(\mathrm{k})\mathbf{q}\right)^2 = \frac{1}{2}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right)^2, \tag{11}$$

where $\mathbf{y}_t = [y(1)\ \cdots\ y(N_t)]^T$. The objective function is minimized by the following equation:

$$\mathbf{y}_t = \mathbf{W}_t\mathbf{p}. \tag{12}$$

The conclusion parameter **p** in Eq. (12) is calculated using the pseudo-inverse of the matrix $\mathbf{W}_t$:

$$\mathbf{p} = \left(\mathbf{W}_t^T\mathbf{W}_t\right)^{-1}\mathbf{W}_t^T\mathbf{y}_t\,, \tag{13}$$

where the matrix $\mathbf{W}_t$ consists of the values of input variable and the normalized values of membership functions, and $\mathbf{y}_t$ consists of target output values.

The DFNN model increases the complexity of the model by passing the predicted value of the very pre-stage FNN module as an input value to the next module and by increasing the number of FNN modules, which may cause the overfitting problem. An algorithm for determining the optimal number of modules to prevent the overfitting problem is applied in the DFNN model. The number of optimal FNN modules is determined and controlled using the values of another fitness function using the verification data:

$$F_v = \exp(-\upsilon_{v1}E_{v1} - \upsilon_{v2}E_{v2}), \tag{14}$$

where

$$E_{v1} = \sqrt{\frac{1}{N_v}\sum_{k=N_t+1}^{N_t+N_v}\left(\frac{y(k)-\hat{y}(k)}{y_{\max}(k)}\right)^2},$$

$$E_{v2} = \max_k\left|\frac{(y(k)-\hat{y}(k))}{y_{\max}(k)}\right|,\ k = N_t+1,\cdots,N_t+N_v,$$

υ_{v1} and υ_{v2} are the weighting values for the RMS and maximum errors, and N_v is the number of verification data. The overall optimization process is depicted in Fig. 5.

The DFNN model is expected to be overfitted when Eq. (15) is satisfied. That is, the fitness function value, F_v, increases rather than decreases when the FNN module is added. If the fitness value of the current FNN module is greater than or equal to that of the very pre-stage FNN module, the output of the current FNN module is more accurate than that of the very pre-stage FNN module; then, the FNN module is added. If the fitness value of the current FNN module is lower than that of the very pre-stage FNN module, the FNN modules are added until that point is determined to be an optimal number.

Table 2
Break size prediction performance obtained in Ref. [2].

Break position	No. of FNN modules	Development data		Test data	
		RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
Hot-leg	8	0.38	1.83	0.51	0.62
Cold-leg	8	0.22	0.78	0.27	0.57
SGT	5	0.77	3.29	0.69	1.58

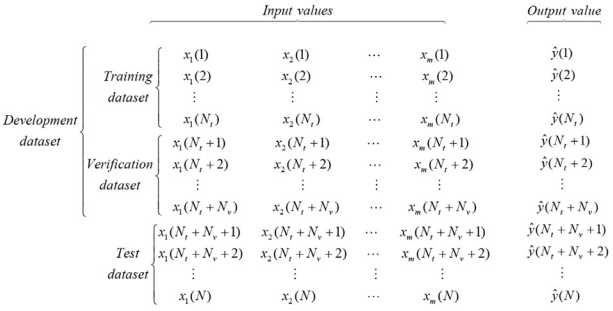


Fig. 7. Data structure to be applied to the DFNN model.

$$g>3 \; and \; [F_v(g) <= 1.1 \times F_v(g-1) \; and \; F_v(g) < F_v(g-3:g-1)], \tag{15}$$

where g is the FNN module number.

Fig. 6 shows that when Eq. (15) is applied to the cold-leg loss of coolant accident (LOCA) for the DFNN model with 13 fuzzy rules, the optimal number of FNN modules is 17 and overfitting occurs at the 18th module.

2.4. Data acquisition for application to the DFNN model

A MAAP code was used to simulate the LOCAs of the OPR-1000 reactor. The DFNN model predicts the output leak flow from the inputs of the elapsed time and LOCA break size. It uses the predicted LOCA break size by utilizing a type of DFNN developed in a previous study [2]. The LOCA break size is estimated by five input values depending on the break position. Tables 1 and 2 list the specific input values and the prediction performance, respectively; the overall RMS error is within 0.7% [2]. Because the predicted break sizes are almost the same as the target size values, they were applied to the DFNN model. The input variables of the DFNN model are the elapsed time and break size for the leak flow estimates. Also, the break position information provided from an algorithm of a previous study [9] was used. Because the sensors may degrade during a severe accident, additional sensor signals were not used. The LOCA break sizes were predicted using sensor signals detected during initial transients (not exposed to severe conditions).

When collecting data, some safety systems (high- and low-pressure safety injection systems) were assumed to be inoperable to simulate a severe post-LOCA circumstance but the passive safety system of safety injection tank (SIT) is available. If the primary pipe breaks, coolant leakage occurs due to the pressure difference between the RCS and the containment. At the beginning of the break, the pressure difference is very large, so the leak flow increases rapidly. It is expected that the injection flow rate into the RCS from SIT increases due to the pressure drop in the RCS accompanying the occurrence of leak flow but the SIT will be empty soon. Simulation data were collected and classified into three accidents: hot-leg LOCA, cold-leg LOCA, and steam generator tube rupture (SGTR). The hot-leg and cold-leg LOCA data were collected from accident simulations with 200 LOCA break sizes at an equal interval from 0.005 to 1 times sectional area inside each pipe. The diameters of the hot-leg and cold-leg were 1.0668 and 0.762 m, respectively. The SGTR data were collected by simulating 1 to 200 SGTRs of tubes with a diameter of 0.0169 m. The data were classified into small and large groups according to their break sizes. The hot-leg and cold-leg LOCAs were divided into 30 simulations for the small group and 170 simulations for the large group, and the SGTR simulations were divided into 100 simulations for the small group and 100 simulations for the large group. The acquired data were divided into training, verification, and test data, as shown in Fig. 7. That is, 100 test data were drawn at fixed intervals from all the data, and training data were selected at the fixed intervals except test data

Table 3
Prediction performance of a single FNN module for hot-leg LOCAs.

(a) Small break size group

No. of fuzzy rules	Development data		Test data	
	RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
5	9.95	59.60	10.77	58.18
10	9.12	46.61	8.31	29.79
20	**8.17**	**45.89**	**8.31**	**26.08**
30	7.32	55.94	8.68	41.06

(b) Large break size group

No. of fuzzy rules	Development data		Test data	
	RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
5	1.19	11.68	0.95	5.16
10	1.68	11.84	1.61	7.17
20	**1.10**	**13.94**	**0.94**	**5.39**
30	0.97	11.11	0.74	5.34

Table 4
Prediction performance of a single FNN module for cold-leg LOCAs.

(a) Small break size group				
No. of fuzzy rules	Development data		Test data	
	RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
5	7.57	41.08	8.34	25.85
10	5.37	52.59	7.89	46.69
20	**6.16**	**75.65**	**4.71**	**25.46**
30	5.74	73.63	6.75	34.61
(b) Large break size group				
No. of fuzzy rules	Development data		Test data	
	RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
5	1.17	5.88	1.17	3.87
10	0.86	6.36	0.81	3.49
20	**0.67**	**5.16**	**0.68**	**3.13**
30	0.57	5.99	0.62	2.94

Table 5
Prediction performance of a single FNN module for SGTRs.

(a) Small break size group				
No. of fuzzy rules	Development data		Test data	
	RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
5	6.08	46.84	6.25	15.27
10	5.67	48.87	5.35	20.09
20	**4.50**	**38.65**	**6.60**	**41.89**
30	3.98	42.10	4.51	14.12
(b) Large break size group				
No. of fuzzy rules	Development data		Test data	
	RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
5	3.52	57.84	1.64	6.66
10	3.41	55.77	1.76	9.38
20	**2.98**	**60.75**	**1.48**	**7.95**
30	3.03	58.39	1.40	5.55

Table 6
Prediction performance of DFNN model for hot-leg LOCAs.

(a) Small break size group					
No. of fuzzy rules	No. of FNN modules	Development data		Test data	
		RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
2	23	0.1885	3.7741	0.0448	0.1388
3	27	0.1695	3.7605	0.0532	0.1487
5	7	0.3172	3.9265	0.0412	0.1654
7	29	0.1469	3.0101	0.0384	0.1788
9	14	0.1675	3.7516	0.0308	0.1544
11	43	0.1432	3.4019	0.0586	0.5712
13	**28**	**0.1054**	**3.0909**	**0.0397**	**0.2134**
15	32	0.1486	3.0852	0.0475	0.4279
(b) Large break size group					
No. of fuzzy rules	No. of FNN modules	Development data		Test data	
		RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
2	34	0.0115	0.2483	0.0069	0.0373
3	37	0.0115	0.2457	0.0184	0.1362
5	19	0.0086	0.2154	0.0101	0.0690
7	26	0.0100	0.3247	0.0121	0.0765
9	15	0.0110	0.2612	0.0152	0.1444
11	15	0.0109	0.2654	0.0134	0.1037
13	**10**	**0.0128**	**0.2831**	**0.0161**	**0.1467**
15	25	0.0086	0.2328	0.0059	0.0454

Table 7
Prediction performance of DFNN model for cold-leg LOCA.

(a) Small break size group

No. of fuzzy rules	No. of FNN modules	Development data		Test data	
		RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
2	10	0.4756	4.7012	0.3428	1.6504
3	26	0.2319	4.1830	0.2088	1.7419
5	14	0.3153	4.5729	0.2030	1.7784
7	23	0.2493	4.4076	0.9088	8.8868
9	11	0.2537	3.2509	0.2173	1.7162
11	8	0.3305	6.5458	0.2149	1.7361
13	**17**	**0.2203**	**4.2426**	**0.1815**	**1.6806**
15	5	0.4269	13.4263	0.2362	1.7003

(b) Large break size group

No. of fuzzy rules	No. of FNN modules	Development data		Test data	
		RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
2	27	0.0337	0.5514	0.0671	0.5024
3	27	0.0277	0.4927	0.0464	0.2628
5	26	0.0270	0.4709	0.0512	0.3334
7	16	0.0330	0.5449	0.0752	0.4899
9	23	0.0248	0.4130	0.0623	0.4240
11	18	0.0265	0.4951	0.0440	0.2458
13	**17**	**0.0271**	**0.5547**	**0.0593**	**0.4659**
15	7	0.0332	0.5394	0.0497	0.2634

Table 8
Prediction performance of DFNN model for SGTRs.

(a) Small break size group

No. of fuzzy rules	No. of FNN modules	Development data		Test data	
		RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
2	19	1.4790	58.6504	0.6218	4.3034
3	11	1.6675	59.4511	1.0749	9.8693
5	15	1.3324	58.7966	0.5270	2.1867
7	27	1.1658	41.6541	0.3631	1.6037
9	19	1.3611	48.1994	0.8915	6.7938
11	5	1.9629	59.1849	0.5903	3.7468
13	**14**	**1.4282**	**41.1940**	**0.3816**	**1.6699**
15	6	1.7054	59.3115	0.6249	3.9665

(b) Large break size group

No. of fuzzy rules	No. of FNN modules	Development data		Test data	
		RMS Error (%)	Maximum Error (%)	RMS Error (%)	Maximum Error (%)
2	4	0.7753	40.7811	0.2534	1.6908
3	4	0.7233	40.5468	0.3912	2.4340
5	4	0.7445	37.4098	0.3568	2.7861
7	4	0.6758	38.2729	0.3171	2.0905
9	4	0.7544	36.3430	0.2925	2.1454
11	4	0.6944	39.6445	0.2761	1.9062
13	**4**	**0.7052**	**35.0808**	**0.2181**	**1.2509**
15	4	0.5439	29.1490	0.1924	1.3104

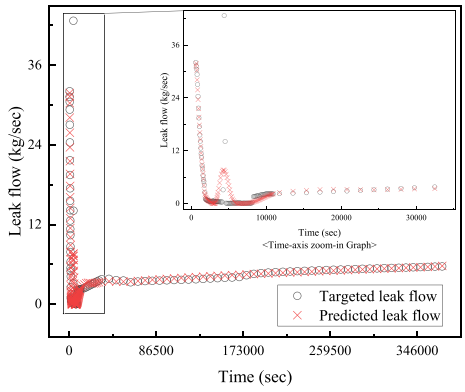
and the remainder was verification data (5% of all data).

3. Prediction results for leak flow using DFNN model

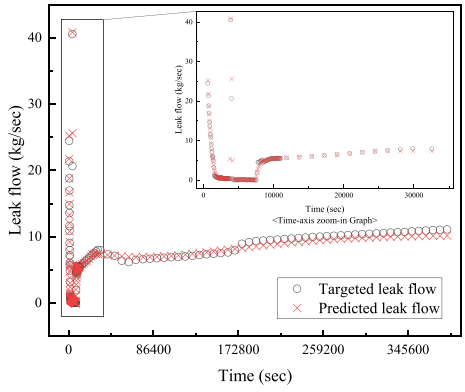
In this study, the leak flow was predicted using the DFNN model. The DFNN with a single FNN module is the same as the FNN model developed in a previous study [4]. The performance of the FNN model was compared to that of the DFNN model. The break size was used as the input of the FNN and DFNN models.

3.1. Leak flow prediction results using the DFNN model and performance comparison with a single FNN module

Tables 3–5 show the leak flow prediction performances of the FNN model (that is, DFNN with a single FNN module) for the hot-leg and cold-leg LOCAs, and the SGTR data, including the RMS and maximum errors according to the number of fuzzy rules. The simulation data of the small break size group fluctuate significantly, so, based on the RMS error of the test data, the prediction

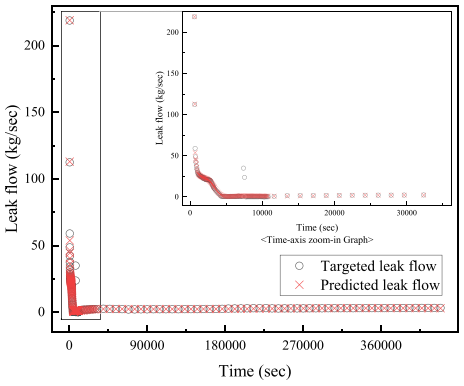


(a) a specific size in the small break size group

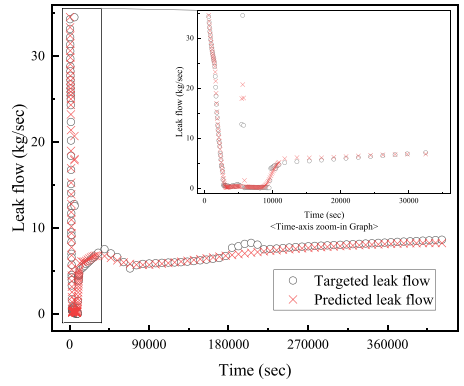


(b) a specific size in the large break size group

Fig. 8. Prediction result of the DFNN model for a hot-leg LOCA with a specific break size (test data of fixed interval): (a) A specific size in the small break size group; (b) A specific size in the large break size group.



(a) a specific size in the small break size group



(b) a specific size in the large break size group

Fig. 9. Prediction result of the DFNN model for a cold-leg LOCA with a specific break size (test data of fixed interval): (a) A specific size in the small break size group; (b) A specific size in the large break size group.

performance is low. However, the large break size groups with small changes show the prediction performance within 2%. It is noticed that the proposed DFNN model predicts the leak flowrate from 600 s after reactor trip due to the LOCA accidents because the DFNN model induces much error due to initial significant transients.

The performance of the DFNN was improved by adding multiple FNN modules in this study. The fuzzy rule number is a hyper-parameter that determines the performance of a DFNN model. Tables 6–8 summarize the prediction performance according to the fuzzy rule number. To determine the optimal fuzzy rule number, the model complexity and RMS error of the test data were chosen as the optimal value selection condition.

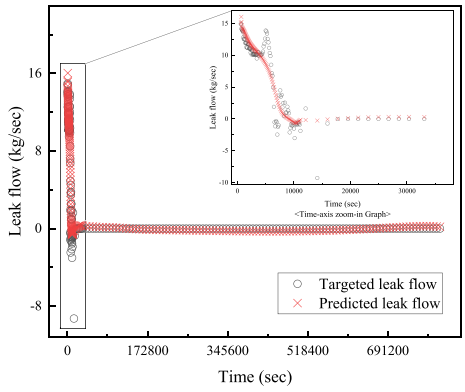
The performance of the DFNN model is significantly higher than when using a single FNN module, particularly in the small break size group. It is clearly shown from comparison of Tables 3–5 and Tables 6–8. Based on model complexity and RMS error, the optimal rule number was chosen to be 13. As the fuzzy rule number increases, the probability of overfitting also increases; thus, the choice of a suitable rule number is important. In all the break cases, the prediction performance is within 0.4% in an aspect of RMS error for the test data. The complexity of the DFNN model was defined as the number of all the conclusion parameters contained in all the FNN modules:

$$\text{model complexity} = (m+1)n + (G-1)(m+1)n, \tag{16}$$

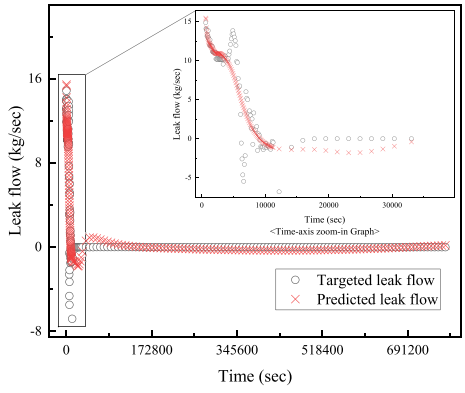
where G is the number of FNN modules, m is the number of input variables to the first FNN module, and n is the number of fuzzy rules. Figs. 8–10 show the results of the DFNN with 13 fuzzy rules for a specific break size in each break position. It is natural that the leak flowrate of large breaks is much larger than that of small breaks in an initial stage (before 600 s have passed). However, as shown in Fig. 9, the leak flowrate of large breaks can be smaller than that of small breaks from 600 s after the reactor trip due to the accidents. When SGTR occurs, RCS coolant from the primary side leaks to the secondary side. With this leakage, the pressure and water level on the primary side decrease, and as time passes, the pressure on the primary side may become lower than the pressure on the secondary side. Due to this pressure reversal, water from the secondary side may flow back to the primary side (refer to Fig. 10).

4. Conclusions

When LOCAs occur in NPPs, many abnormal signals are detected. Currently, it is important for operators to understand the abnormal signals and take proper actions. However, during LOCAs, these signals change rapidly, increasing the workload and the



(a) a specific size in the small break size group



(b) a specific size in the large break size group

Fig. 10. Prediction result of the DFNN model for an SGTR with a specific break size (test data of fixed interval): (a) A specific size in the small break size group; (b) A specific size in the large break size group.

probability of human errors. In this study, we attempted to reduce the human errors of inappropriate actions by providing a leak flow prediction signal in case of LOCAs. The leak flow was predicted with a DFNN model using the LOCA break size predicted in the previous study [2] as input data. In addition, the least-squares method and a GA were applied to optimize the DFNN model, and also another fitness function in the GA was introduced to determine the optimal number of the FNN modules. The fuzzy rule number of the developed DFNN model varied depending on the break position, and, considering the RMS error and model complexity, the optimal fuzzy rule number was determined to be 13. The RMS errors of the leakage amount in the small and large hot-leg break size groups predicted from the DFNN model with 13 fuzzy rules were found to be 0.0397% and 0.0161%, respectively, those of the small and large cold-leg break size groups were 0.1815% and 0.0593%, respectively, and those of the small and large break groups for the SGTRs were found to be 0.3816% and 0.2181%, respectively. These results show that the performance is improved to 0.3816%, which is the largest RMS error of the DFNN model, compared with 8.31%, which is the largest RMS error of a single FNN module (FNN model) for the entire break positions. The DFNN model showed a significantly higher performance than a single FNN module (FNN model), within a 0.4% RMS error. The DFNN model is expected to help operators to take proper safety actions by providing them with important information of leak flows in the LOCAs of NPPs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (Grant No. NRF-2018M2B2B1065651) and the Korea Institute of Energy Technology and Planning (KETEP) grant funded by the Korean government (MOTIE) (Grant No. 20181510102340, Development of a real-time detection system for unidentified RCS leakage less than 0.5 gpm).

References

[1] "Operational Performance Information System for Nuclear Power Plant", Nuclear Accident and Failure Status, 2019 last modified Sep. 06, http://opis.kins.re.kr/opis?act=KROBA3100R. (Accessed 30 December 2019).
[2] G.P. Choi, K.H. Yoo, J.H. Back, M.G. Na, Estimation of LOCA break size using cascaded fuzzy neural networks, Nucl. Eng. Tech. 49 (2017) 495–503.
[3] Y.J. An, K.H. Yoo, M.G. Na, Y.S. Kim, Critical flow prediction using simplified cascade fuzzy neural networks, Ann. Nucl. Energy 136 (2020) 107047.
[4] D.Y. Kim, K.H. Yoo, J.H. Kim, M.G. Na, S. Hur, C.H. Kim, Prediction of leak flow rate using fuzzy neural networks in severe post-LOCA circumstances, IEEE Trans. Nucl. Sci. 61 (2014) 3644–3652.
[5] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man Mach. Stud. 7 (1975) 1–13.
[6] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Systems, Man, Cybern. SMC 1 (1985) 116–132.
[7] D.Y. Kim, K.H. Yoo, M.G. Na, Estimation of minimum DNBR using cascaded fuzzy neural networks, IEEE Trans. Nucl. Sci. 62 (2015) 1849–1856.
[8] D.Y. Kim, K.H. Yoo, G.P. Choi, J.H. Back, M.G. Na, Reactor vessel water level estimation during severe accidents using cascaded fuzzy neural networks, Nucl. Eng. Technol. 48 (2016) 702–710.
[9] K.H. Yoo, Y.D. Koo, J.H. Back, M.G. Na, Identification of LOCA and estimation of its break size by multi-connected support vector machines, IEEE Trans. Nucl. Sci. 64 (10) (2017) 2610–2617.