Available online at ScienceDirect

Nuclear Engineering and Technology

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

Original Article

Estimation of LOCA Break Size Using Cascaded Fuzzy Neural Networks

Geon Pil Choi, Kwae Hwan Yoo, Ju Hyun Back, and Man Gyun Na *

Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju 501-759, Republic of Korea

ARTICLE INFO

Article history:
Received 22 July 2016
Received in revised form
10 October 2016
Accepted 4 November 2016
Available online 16 November 2016

Keywords:
Artificial Intelligence
Cascaded Fuzzy Neural Networks
Genetic Algorithm
LOCA Break Size
Loss-of-coolant Accident (LOCA)

ABSTRACT

Operators of nuclear power plants may not be equipped with sufficient information during a loss-of-coolant accident (LOCA), which can be fatal, or they may not have sufficient time to analyze the information they do have, even if this information is adequate. It is not easy to predict the progression of LOCAs in nuclear power plants. Therefore, accurate information on the LOCA break position and size should be provided to efficiently manage the accident. In this paper, the LOCA break size is predicted using a cascaded fuzzy neural network (CFNN) model. The input data of the CFNN model are the time-integrated values of each measurement signal for an initial short-time interval after a reactor scram. The training of the CFNN model is accomplished by a hybrid method combined with a genetic algorithm and a least squares method. As a result, LOCA break size is estimated exactly by the proposed CFNN model.

1. Introduction

Because of continuously increasing energy demands, many nuclear power plants (NPPs) are in operation globally. Many of these NPPs have been in long-term operation, and may be slightly more vulnerable to accidents, such as loss-of-coolant accidents (LOCAs), because the pipes in these plants can be weak. NPPs automatically operate emergency core cooling systems, such as safety injection systems (SISs), when a LOCA occurs. The emergency core cooling systems might not function properly in case of a LOCA with a small break size, due to high pressure keeping in the pipes. Additionally, when an accident occurs in an NPP, the plant operators may have only incomplete information or may not have sufficient time to analyze the accident even though they are provided enough information. In these cases, operators must analyze abnormal plant conditions using temporary trends of important parameters from the time of the accident's occurrence. However, they may have difficulty in predicting the accident looking at the displayed temporary trend of parameters accessible from the main control room [1]. If the operators were offered information on the LOCA break size immediately after the LOCA, they could minimize the damage caused by the LOCA. Therefore, this study estimates the LOCA break size.

A number of artificial intelligence techniques characterized as machine learning has been applied successfully to many nuclear engineering areas, such as signal validation [2–4], plant diagnostics [5–8], and smart sensing (or function approximation) [9–11]. In this study, a cascaded fuzzy neural network (CFNN) model with a machine learning function was utilized to predict the LOCA break size. The CFNN model predicts the LOCA break size by a repetitively performed analysis

* *Corresponding author.*
E-mail address: magyna@chosun.ac.kr (M.G. Na).
http://dx.doi.org/10.1016/j.net.2016.11.001

using continuously connected fuzzy neural network (FNN) modules. In effect, the CFNN is an extension of the FNN [12].

The CFNN model based on artificial intelligence requires acquired data for its development and verification. Because a variety of real LOCA accident data cannot be obtained from actual NPP accidents, the data used herein were obtained by numerically simulating severe accident scenarios of an optimized power reactor (OPR1000) using Modular Accident Analysis Program (MAAP) code (Fauske & Associates, Burr Ridge, IL, USA).

2. CFNN methodology

2.1. CFNN model

The CFNN model consists of more than two FNN modules, of which each stage corresponds with a single-stage FNN module. The FNN model is a combination of a fuzzy inference system and neuronal training. In the usual fuzzy inference system that is called the Mamdani fuzzy model [13], the "if" part is fuzzy linguistic and the "then" part is fuzzy linguistic, too, which requires a defuzzification process since the LOCA break size estimation problem at hand has the input and output of real values. In this study, a Takagi–Sugeno type fuzzy model is used in which the "if" part is fuzzy linguistic, while the "then" part is crisp [14]. Therefore, the Takagi–Sugeno type fuzzy inference system does not need a defuzzifier in the output terminal. The Takagi–Sugeno type fuzzy inference system can be described as follows:

$$\begin{aligned} &If\ s_1(t)\ is\ A_{i1}\ AND\cdots AND\ s_m(t)\ is\ A_{im},\ then \\ &\tilde{y}_i = f_i[s_1(t),L,s_m(t)] \end{aligned} \tag{1}$$

Most studies using the FNN models have suggested different types of single-stage fuzzy reasoning mechanisms. However, single-stage fuzzy reasoning is the simplest among the various human reasoning mechanisms. Syllogistic fuzzy reasoning, where the results of a rule in preceding stage is passed to the current stage as a fact, could effectively accumulate a grand-scale system with high-level knowledge [15]. Because the CFNN model is expected to offer a better performance than a simple FNN model, this study used a CFNN model based on syllogistic fuzzy reasoning. Eq. (2) shows the Takagi–Sugeno-type fuzzy inference system of the CFNN model.

$$\begin{aligned}
&\text{Stage 1}\ \begin{bmatrix} If\ s_1(t)\ is\ A^1_{i1}(t)\ AND\cdots AND\ s_m(t)\ is\ A^1_{im}(t), \\ then\ \tilde{y}^1_i(t) = f^1_i[s_1(t),\cdots,s_m(t)] \end{bmatrix} \\
&\text{Stage 2}\ \begin{bmatrix} If\ s_1(t)\ is\ A^2_{i1}(t)\ AND\cdots AND\ s_m(t)\ is\ A^2_{im}(t) \\ \quad AND\ \tilde{y}^1(t)\ is\ A^2_{i(m+1)}(t), \\ then\ \tilde{y}^2_i(t) = f^2_i[s_1(t),\cdots,s_m(t),\tilde{y}^1(t)] \end{bmatrix} \\
&\qquad\vdots \\
&\text{Stage } l\ \begin{bmatrix} If\ s_1(t)\ is\ A^l_{i1}(t)\ AND\cdots AND\ s_m(t)\ is\ A^l_{im}(t), \\ \quad AND\ \tilde{y}^1(t)\ is\ A^l_{i(m+1)}(t)\ AND\cdots AND\ \tilde{y}^{l-1}(t)\ is\ A^l_{i(m+l-1)}(t), \\ then\ \tilde{y}^l_i(t) = f^l_i[s_1(t),\cdots,s_m(t),\tilde{y}^1(t),\cdots,\tilde{y}^{l-1}(t)] \end{bmatrix} \\
&\frac{\text{Fact}: s_1(t)\ is\ A^1_{i1}(t)\ AND\cdots AND\ s_m(t)\ is\ A^1_{im}(t)}{\text{Consequent}: \tilde{y}^l_i(t) = f^l_i[s_1(t),\cdots,s_m(t),\tilde{y}^1(t),\cdots,\tilde{y}^{l-1}(t)]}
\end{aligned} \tag{2}$$

where $s_j(t)$: FIS input value ($j=1,2,\cdots,m$); $A^k_{ij}(t)$: fuzzy set for the i^{th} fuzzy rule ($i=1,2,\cdots,n$) and the j^{th} input variable at the k^{th} stage ($k=1,2,\cdots,l$); $\tilde{y}^k_i(t)$: i^{th} fuzzy rule output at the k^{th} stage; $\tilde{y}^k(t)$: CFNN model output at the k^{th} FNN module; l: number of FNN modules; m: number of input variables; and n: number of fuzzy rules.

The number of input and output training data, T, of the fuzzy model in Eq. (3) are assumed to be available, and each data point is assumed to be a normalized value.

$$\mathbf{c}^T(t) = [\mathbf{s}^T(t), \tilde{y}(t)] \tag{3}$$

where

$$\mathbf{s}^T(t) = [s_1(t), s_2(t), \cdots, s_m(t)],\quad t = 1,2,\cdots,T.$$

In the function in Eq. (2), the output of an arbitrary i^{th} rule,$f_i[\mathbf{s}(t)]$, is made of the first-order polynomial of inputs as given in Eq. (4).

$$f_i[\mathbf{s}(t)] = \sum_{j=1}^{m} q_{ij} s_j(t) + o_i \tag{4}$$

where q_{ij}: weight of the i^{th} fuzzy rule and j^{th} input variable; and o_i: bias of the i^{th} fuzzy rule.

The CFNN model structure contains serially connected FNN modules. Therefore, only the first FNN module will be explained. The process of the first-stage FNN module is shown in Fig. 1. Each stage is composed of six layers. The first layer is composed of the input nodes that transfer input values to the

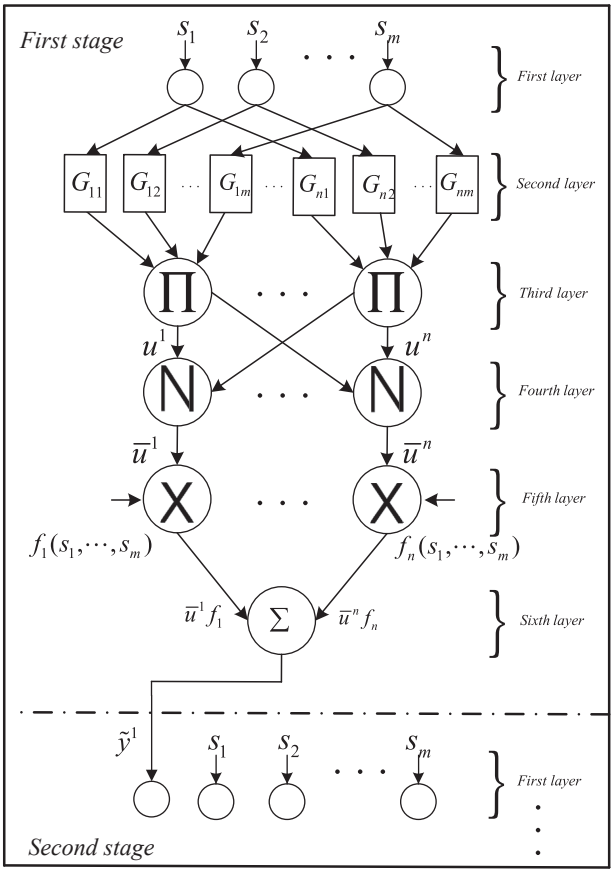

Fig. 1 – First stage fuzzy neural network module.

second layer. Each output from the first layer is transferred to the inputs of the membership function. The membership function of fuzzy sets $A_{ij}(t)$ is denoted as $G_{ij}[s_j(t)]$. In this study, the simple symmetric Gaussian membership function in Eq. (5) is applied to lower the number of parameters to be optimized compared with the asymmetric Gaussian membership functions. It has a characteristic symmetric bell curve shape that tends to zero. The second layer is the fuzzification layer, which calculates the membership function values using Eq. (5).

$$G_{ij}[s_j(t)] = e^{-\frac{[s_j(t)-x_{ij}]^2}{2\sigma_{ij}^2}} \tag{5}$$

where x_{ij}: center position of the peak; and σ_{ij}: width of the bell shape.

The third layer in Fig. 1 carries out the fuzzy inference system, and each node in this layer multiplies the membership function values from the second layer and the output of this layer is given by the product as Eq. (6). The fourth layer performs normalization using Eq. (7).

$$u^i[\mathbf{s}(t)] = \prod_{j=1}^{m} G_{ij}[s_j(t)] \tag{6}$$

$$\overline{u}^i(t) = \frac{u^i[\mathbf{s}(t)]}{\sum_{i=1}^{n} u^i[\mathbf{s}(t)]} \tag{7}$$

The fifth layer in Fig. 1 generates the output values of each fuzzy "if–then" rule. Finally, the sixth layer aggregates all the fuzzy "if–then" rules and is expressed as Eq. (8):

$$\tilde{y}(t) = \sum_{i=1}^{n} \overline{u}^i(t) y_i(t) = \sum_{i=1}^{n} \overline{u}^i(t) f_i[\mathbf{s}(t)] = \mathbf{u}^T(t)\mathbf{q} \tag{8}$$

where

$\mathbf{u}(t) = [\overline{u}^1(t)s_1(t)\cdots\overline{u}^n(t)s_1(t)\cdots\cdots\overline{u}^1(t)s_m(t)\cdots\overline{u}^n(t)s_m(t)\ \overline{u}^1(t)\cdots\overline{u}^n(t)]^T$; and $\mathbf{q} = [q_{11}\cdots q_{n1}\cdots\cdots q_{1m}\cdots q_{nm}\ o_1\cdots o_n]^T$.

Vector $\mathbf{q}$ presents a consequent parameter vector with a dimension of $(m+1)\times n$ and vector $\mathbf{u}(t)$ consist of the input data and membership function values. Therefore, the estimated output of T input and output training data induced from Eq. (8) can be expressed as Eq. (9).

$$\tilde{\mathbf{y}}_t = \mathbf{U}_t\mathbf{q} \tag{9}$$

where $\tilde{\mathbf{y}}_t = [\tilde{y}(1)\ \tilde{y}(2)\cdots\tilde{y}(T)]^T$; and $\mathbf{U}_t = [\mathbf{u}(1)\ \mathbf{u}(2)\cdots\mathbf{u}(T)]^T$.

Matrix $\mathbf{U}_t$ has a dimension of $[(m+1)\times n]\times T$.

The second-stage FNN module utilizes the initial input data and the output of the first-stage FNN module as its input variables. This process is repeated l times to find the optimum output value. Fig. 2 presents the architecture of the CFNN model.

2.2. CFNN optimization

The CFNN model that was developed to estimate a LOCA break size is optimized by using the specified training data. The optimization method combines a genetic algorithm and the least squares method. The antecedent parameters in the membership function in Eq. (5) are optimized by a genetic algorithm. The consequent parameters in Eq. (4) are optimized by the least squares method [12]. In the genetic algorithm, the

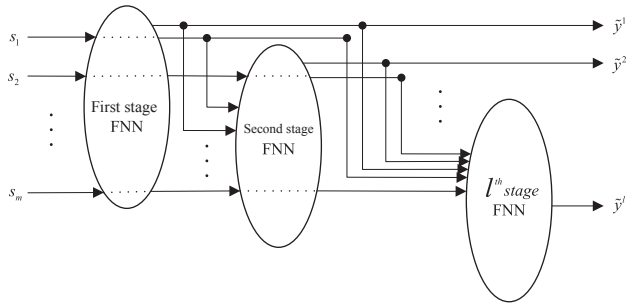

Fig. 2 – Typical diagram of the cascaded fuzzy neural network (FNN) model.

following fitness function is proposed to minimize root-mean-square (RMS) errors:

$$F = \exp[-\lambda(E_t + 2E_v)] \tag{10}$$

where: $E_t = \sqrt{\frac{1}{T}\sum_{t=1}^{T}[y(t)-\tilde{y}(t)]^2}$; $E_v = \sqrt{\frac{1}{V}\sum_{t=T+1}^{T+V}[y(t)-\tilde{y}(t)]^2}$; λ: weight value of the RMS error; T: number of training data; V: number of verification data; $y(t)$: target output value; and $\tilde{y}(t)$: estimated value by CFNN.

As shown in Eq. (10), the RMS term of the verification data is overweighed two times more than that of the training data, to mitigate the overfitting problem. If the antecedent parameters are decided by a genetic algorithm through genetic operations such as selection, crossover, and mutation, the consequent parameters appear similarly to Eq. (9) as first-order combinations. Consequent parameter $\mathbf{q}$ is optimized by the least squares method and is computed to minimize the objective function represented by the squared error between target value $y(t)$ and estimated value $\tilde{y}(t)$ [16].

$$J = \sum_{t=1}^{T}[y(t)-\tilde{y}(t)]^2 = \sum_{t=1}^{T}[y(t)-\mathbf{u}^T(t)\mathbf{q}]^2 = \frac{1}{2}(\mathbf{y}_t-\tilde{\mathbf{y}}_t)^2 \tag{11}$$

where $\mathbf{y}_t = [y(1)\ y(2)\cdots y(T)]^T$.

The solution for minimizing the objective function in Eq. (11) is calculated by the following equation:

$$\mathbf{y}_t = \mathbf{U}_t\mathbf{q} \tag{12}$$

where

$\mathbf{U}_t = [\mathbf{u}(1)\ \mathbf{u}(2)\cdots\mathbf{u}(T)]^T$.

Parameter vector $\mathbf{q}$ in Eq. (11) is solved using the pseudo-inverse as follows:

$$\mathbf{q} = (\mathbf{U}_t^T\mathbf{U}_t)^{-1}\mathbf{U}_t^T\mathbf{y}_t. \tag{13}$$

Parameter vector $\mathbf{q}$ is composed of a series of input data, output data, and their membership function values, because matrix $\mathbf{U}_t$ comprises the input data and membership function values, and $\mathbf{y}_t$ contains the output data. Fig. 3 presents the optimization procedure of the CFNN model. Fig. 4 depicts a data structure used in developing the CFNN model.

The complexity of the CFNN model is supposed to be proportional to the number of elements in parameter vector $\mathbf{q}$ in Eq. (13). Therefore, its complexity is defined as the number of

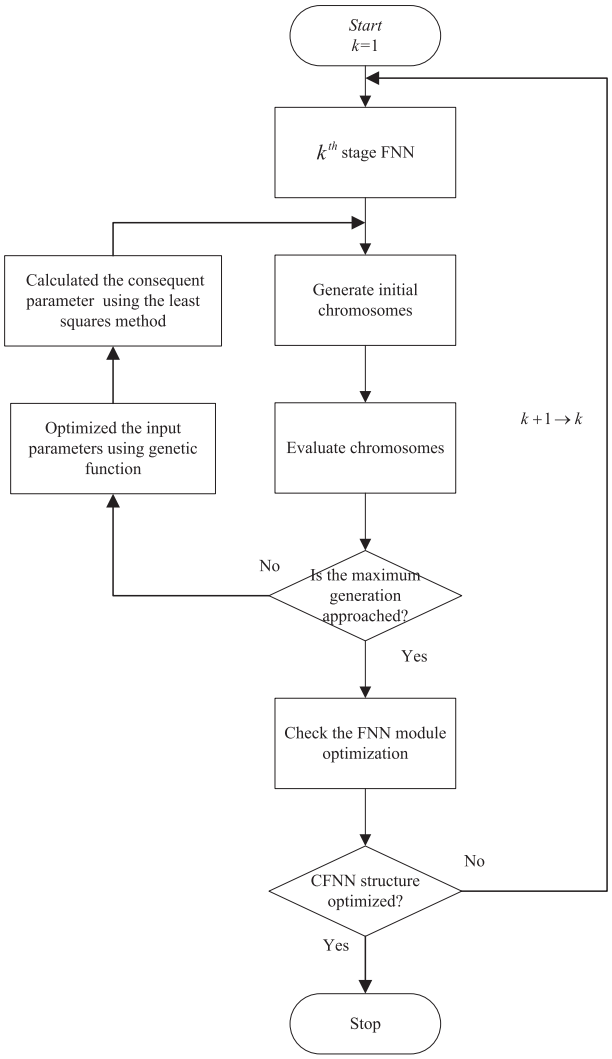

Fig. 3 – Optimization procedure for the cascaded fuzzy neural network (CFNN) model.

elements in parameter vector **q** of all FNN modules contained in the CFNN and is calculated as follows [12]:

$$Complexity = \frac{l(l+2m+1)n}{2} \tag{14}$$

where l: number of FNN modules; m: number of input variables; and n: number of fuzzy rules.

The complexity of the CFNN model radically increases as the number of FNN modules increases and linearly increases according to the number of fuzzy rules.

The CFNN model might encounter problems with overfitting. When an overfitting sign occurs, the process of adding FNN modules will stop. The overfitting problem can be resolved through cross-checking using the data structure shown in Fig. 4 [16]. A criterion used to evaluate whether overfitting has occurred at stage k is expressed as the sum of the errors for the verification data as follows:

$$F(k) = \sum_{t=T+1}^{V} \left[y(t) - \tilde{y}^k(t)\right]^2. \tag{15}$$

The training and checking processes stop if $F(k+1) > F(k)$ (refer to Fig. 5), which means that the errors of the verification data increase according to the increase in the number of stages. When the condition $[F(k+1) > F(k)]$ is satisfied, the CFNN model may begin to become overfit if the process of adding FNN modules continues. If this condition is not satisfied, the algorithm moves to the next stage, and an FNN module is added. Further, the complexity defined in Eq. (14) that is proportional to the square of the number of stages should be smaller than the number of training data to prevent the potential ill-posed problem related to the pseudoinverse in Eq. (13). The CFNN model with l FNN modules that satisfy these two conditions is drawn in Fig. 2.

3. Estimation of the LOCA break size

The used data were obtained by simulating the MAAP code for the LOCA scenarios of the OPR1000 that is a pressurized water reactor developed in Korea. It is plain that the MAAP code will have inaccuracies in the simulation results. In order to assess

Output	Inputs				
$y(1)$	$s_1(1)$	$s_2(1)$	$\cdots$	$s_m(1)$	Training dataset
$y(2)$	$s_1(2)$	$s_2(2)$	$\cdots$	$s_m(2)$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
$y(T)$	$s_1(T)$	$s_2(T)$	$\cdots$	$s_m(T)$	
$y(T+1)$	$s_1(T+1)$	$s_2(T+1)$	$\cdots$	$s_m(T+1)$	Development dataset
$y(T+2)$	$s_1(T+2)$	$s_2(T+2)$	$\cdots$	$s_m(T+2)$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	Verification dataset
$y(T+V)$	$s_1(T+V)$	$s_2(T+V)$	$\cdots$	$s_m(T+V)$	
$y(T+V+1)$	$s_1(T+V+1)$	$s_2(T+V+1)$	$\cdots$	$s_m(T+V+1)$	
$y(T+V+2)$	$s_1(T+V+2)$	$s_2(T+V+2)$	$\cdots$	$s_m(T+V+2)$	Test dataset
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
$y(N)$	$s_1(N)$	$s_2(N)$	$\cdots$	$s_m(N)$	

Fig. 4 – Data structure for developing and testing the cascaded fuzzy neural network (CFNN) model.

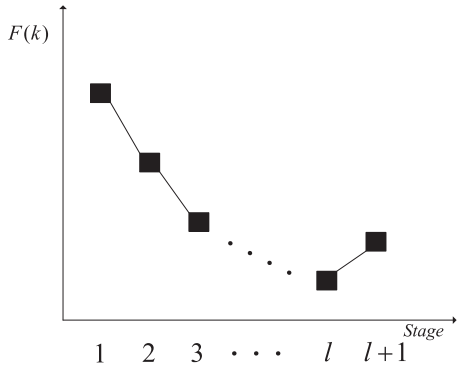

Fig. 5 – Fractional error $F(k)$ according to stage number.

this inaccuracy, many researchers have analyzed the simulation results of MAAP code including other thermal hydraulic codes. Lindholm et al. [17] examined core refloodings with three severe accident analysis computer codes, such as MAAP, MELCOR, and SCDAP/RELAP5, and reported that all the three codes predicted similar trends with relation to the thermal hydraulic phenomena at the reflood phase. Allison [18] also compared the simulation results of MAAP with results using MELCOR and SCDAP/RELAP5 for large break LOCA events. This study showed similar results in the early phase, even though user models influenced the simulation results in a later phase.

The LOCA break position could not be detected. Therefore, the break position needs to be identified and predicted. In previous studies [1,19], the LOCA break positions were accurately identified. These simulations comprised 600 cases of severe accident scenarios. The data consisted of 200 hot-leg LOCAs, 200 cold-leg LOCAs, and 200 steam generator tube

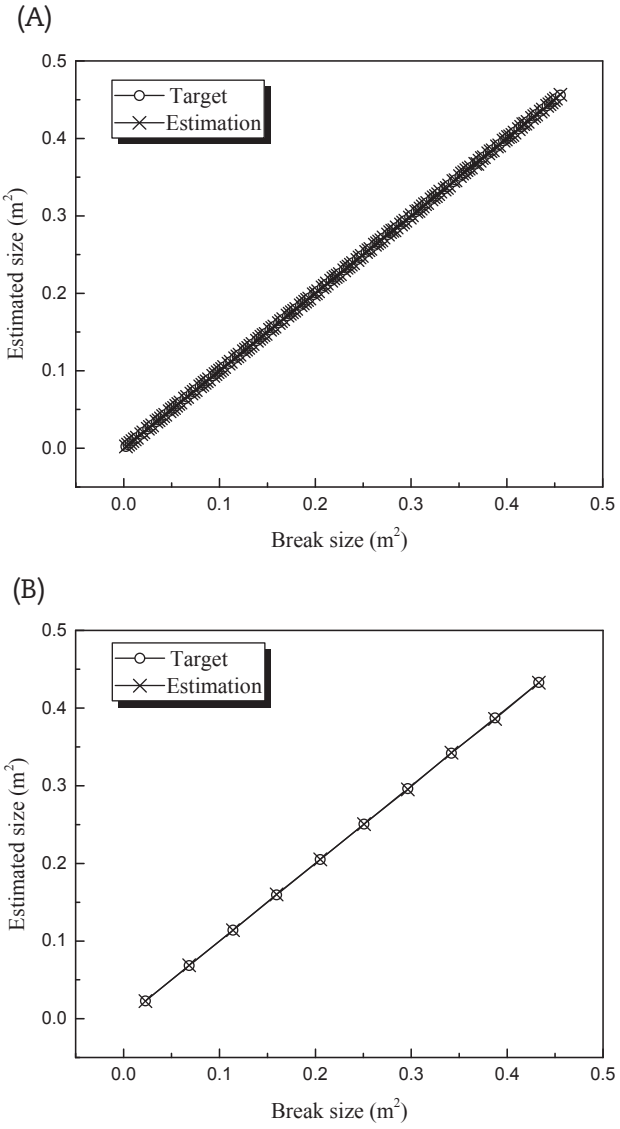

Fig. 6 – Estimation performance of the cascaded fuzzy neural network model for hot-leg loss-of-coolant accidents. (A) Estimated break size for development data. (B) Estimated break size for test data.

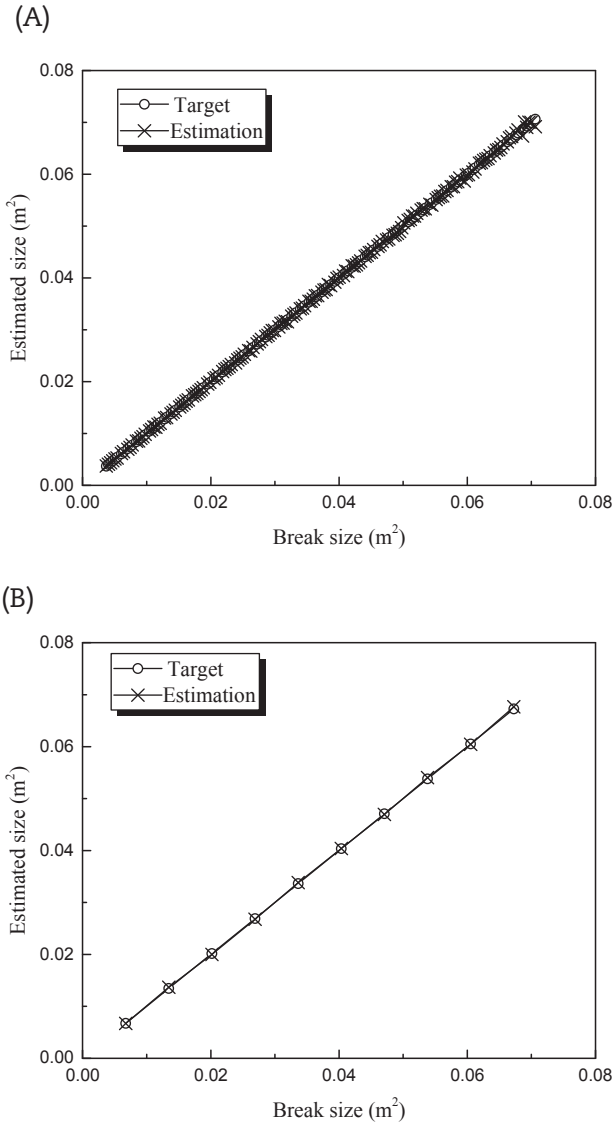

Fig. 7 – Estimation performance of the cascaded fuzzy neural network model for cold-leg loss-of-coolant accidents. (A) Estimated break size for development data. (B) Estimated break size for test data.

ruptures (SGTRs). The 600 accident simulations are divided into development simulation data and test simulation data in Fig. 4. The development data are used to devise and optimize the CFNN model, and the test data are used to independently verify the CFNN model. Therefore, a total of 570 simulations for development of the CFNN model is composed of each 190 simulations at the hot-leg LOCA, cold-leg LOCA, and SGTR. The remaining 30 simulations for verifying the CFNN model consist of each 10 simulations at the hot-leg LOCA, cold-leg LOCA, and SGTR, and are used as test data. The test data were picked at a fixed interval when the acquired data was sorted according to the LOCA break size. Information on the LOCA break size of the selected test data is shown in Figs. 6B, 7B, and 8B.

The accidents have different break positions and sizes. The inner diameters of the hot-leg and cold-leg pipes are 1.0668 m and 0.762 m, respectively, and the inner diameter of a steam generator tube is 0.0169 m. The break size of the hot-leg and cold-leg pipes ranges from a minimum of 1/400 of their guillotine break to a maximum of half of the guillotine break. The break size of SGTR ranges from 11 SGTRs to 210 SGTRs. The data were recognized by the simulated sensor signals that are collected from these simulations and compose a total of 13 signals: temperature of core exit, pressure and temperature in containment, pressure and water level in pressurizer, sump water level, reactor pressure vessel water level, pressure, and temperature in broken side steam generator (S/G), water level in broken side S/G, pressure and temperature in the unbroken side S/G, and water level in the unbroken side S/G. The pressure and temperature in containment are computed values at a center position of containment that is known as an upper compartment below the containment dome. OPR1000 has two S/Gs. The terms "broken side S/G" and "unbroken side S/G" conform to the two S/Gs that are connected to the broken hot-leg (or cold-leg or SGTs) and the unbroken hot-leg (or cold-leg or SGTs), respectively [1]. The input data variables to the CFNN are the time-integrated values of 13 simulated sensor signals as follows:

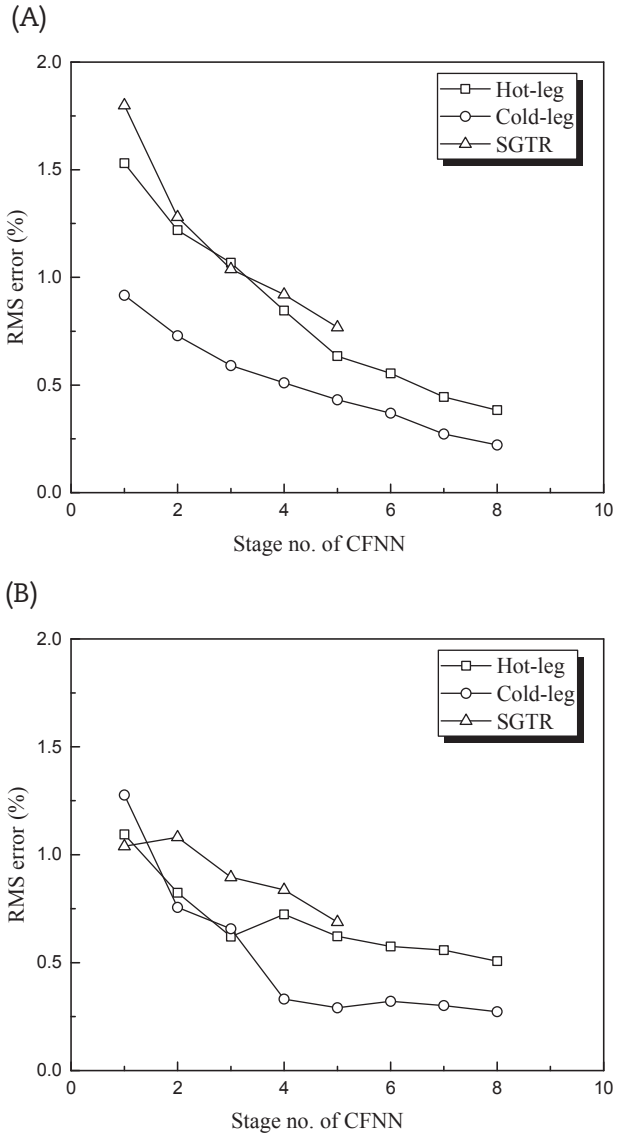

Fig. 8 – Estimation performance of the cascaded fuzzy neural network model for steam generator tube ruptures. (A) Estimated break size for development data. (B) Estimated break size for test data.

$$x_j = \int_{t_s}^{t_s+\Delta t} g_j(t)dt, \quad j = 1, 2, \cdots, 13 \tag{16}$$

where $y_j(t)$: specific simulated sensor signal; t_s: scram time point; and Δt: integration time span.

The integration time span in Eq. (16) is 60 s, which means that the CFNN uses the time-integrated signals over a 60-second time interval immediately after the reactor scram. Since the estimation error of the CFNN was only a little sensitive to the integration time span, the integration time span was determined by considering the estimation error. Also, because a variety of transients can be induced due to safety system actuation if the integration time is long, the integration time span was decided to be as short as possible so that the transients of the used signals are affected as little as possible.

The CFNN model did not use all of the 13 acquired signals. All data (13 signals) trend differently as a result of the initiating events having different break sizes, which means that any of them can be selected as an input for the CFNN model to estimate the LOCA break size. The input data are selected by considering the correlation between the time-integrated values of the simulated sensor signals and the break size. The sensor signals with high correlation with the output data (break size) are selected as the input data. The CFNN model was

Table 1 – Input signals for estimating the loss-of-coolant accidents break size using a cascaded fuzzy neural network.

Break position	Simulated input signals
Hot-leg Cold-leg	Pressure in containment, temperature in containment, pressure in pressurizer, water level in pressurizer, pressure in a broken side S/G
SGT	Temperature in containment, RPV water level, water level in a broken side S/G, pressure 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.

Table 2 – Performance of the cascaded fuzzy neural network model.

Break position	No. of FNN modules	Complexity	Development data		Test data	
			RMS error (%)	Maximum error (%)	RMS error (%)	Maximum error (%)
Hot-leg	8	152	0.38	1.83	0.51	0.62
Cold-leg	8	152	0.22	0.78	0.27	0.57
SGT	5	80	0.77	3.29	0.69	1.58

FNN, fuzzy neural network; RMS, root-mean-square.

optimized by a genetic algorithm and the least squares method using the development data. The parameter values that are related to the genetic algorithm were as follows: the crossover probability was 100%, the mutation probability was 5%, and the population size was 30. Also, the number of fuzzy rules is 2.

Table 1 shows the selected sensor signals to estimate the LOCA break size. The number of selected input data is five for each break position. For hot-leg and cold-leg LOCAs, the selected input data are pressure and temperature in containment, pressure and water level in pressurizer, and pressure in a broken side S/G. Additionally, for SGTR, the selected input data are temperature in containment, reactor pressure vessel water level, water level, and pressure in a broken side S/G, and temperature in an unbroken side S/G.

Table 2 lists the performance results achieved by the CFNN model for hot-leg LOCAs, cold-leg LOCAs, and SGTRs. As listed in Table 2, for the development data of the hot-leg LOCAs, the RMS error and maximum error were approximately 0.38% and 1.83%, respectively. For the development data of the cold-leg LOCAs, the RMS error and maximum error are approximately 0.22% and 0.78%, respectively. For the development data of SGTRs, the RMS error and maximum error are approximately 0.77% and 3.29%, respectively.

As listed in Table 2, for the test data of the hot-leg LOCAs, the RMS error and maximum error were approximately 0.51% and 0.62%, respectively. For the test data of the cold-leg LOCAs, the RMS error and maximum error are approximately 0.27% and 0.57%, respectively. For the test data of SGTRs, the RMS error and maximum error are approximately 0.69% and 1.58%, respectively. Table 3 lists the RMS and maximum errors in the test data for the CFNN and support vector regression (SVR) model [20]. The SVR model nonlinearly maps the original data into higher dimensional feature space and conducts linear regression on the resulting feature space. The kernel used in the SVR model uses the radial basis function. Three SVR models were developed for the three different LOCA positions. The kernel parameters of the SVR models were optimized by a genetic algorithm like the proposed CFNN model. The CFNN and SVR models used the same input data. The CFNN model performs much better than the SVR model does.

Table 3 – Comparison of the cascaded fuzzy neural network (CFNN) and support vector regression (SVR) models.

Break position	CFNN		SVR	
	RMS error (%)	Maximum error (%)	RMS error (%)	Maximum error (%)
Hot-leg	0.51	0.62	1.00	2.89
Cold-leg	0.27	0.57	0.67	1.91
SGT	0.69	1.58	0.64	1.45

RMS, root-mean-square; SGT, steam generator tube.

Figs. 6–8 show the estimated break sizes for the development and test data for the hot-leg LOCAs, cold-leg LOCAs, and SGTRs, respectively. The estimation values are consistent with the target values for hot-leg LOCAs, cold-leg LOCAs, and SGTRs, respectively. The CFNN model is confirmed to accurately estimate the LOCA break size. Fig. 9 shows the RMS

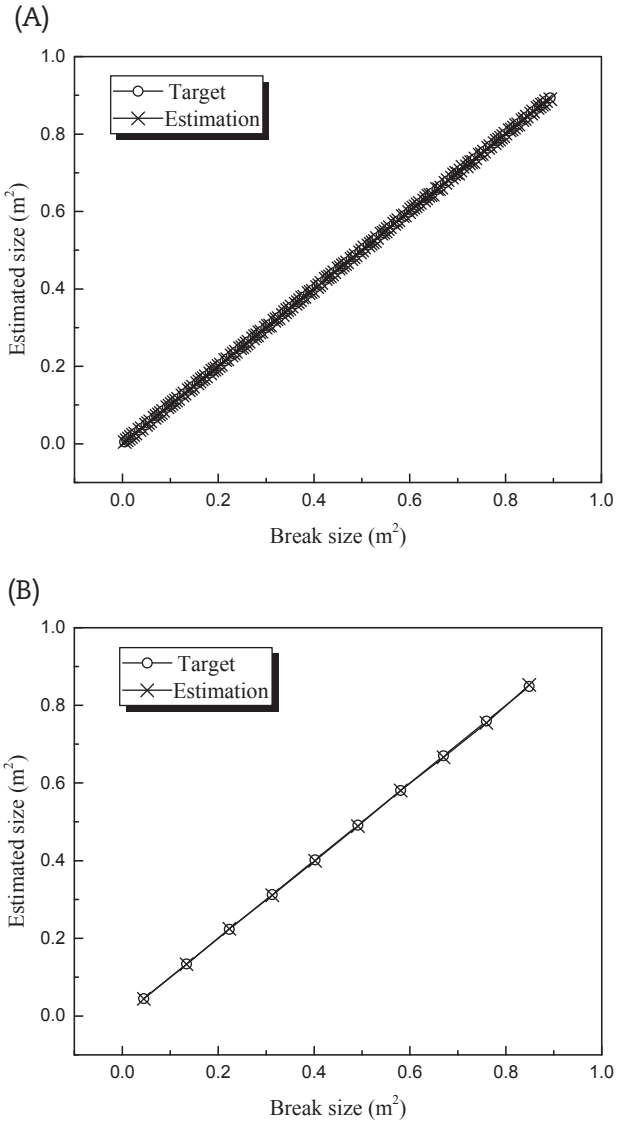

Fig. 9 – RMS error versus stage number. (A) Development data. (B) Test data. CFNN, cascaded fuzzy neural network; RMS, root-mean-square; SGTR, steam generator tube ruptures.

Table 4 – Performance degradation of the cascaded fuzzy neural network model due to biased measurement uncertainties.

	Hot-leg LOCA					Cold-leg LOCA					SGTR				
	Measurement error (%)					Measurement error (%)					Measurement error (%)				
	0.0	+1.0	−1.0	+2.0	−2.0	0.0	+1.0	−1.0	+2.0	−2.0	0.0	+1.0	−1.0	+2.0	−2.0
RMS error (%)	0.51	3.34	3.42	7.35	5.88	0.27	2.62	2.60	5.39	5.04	0.69	1.15	1.32	2.39	2.22
Maximum error (%)	0.62	4.73	5.41	11.13	9.72	0.57	4.47	4.42	9.22	8.54	1.58	2.15	3.16	4.17	4.51

LOCA, loss-of-coolant accidents; RMS, root-mean-square; SGTR, steam generator tube ruptures.

errors of the development and test data, respectively. The RMS error gradually decreases as the number of stages in the CFNN model increases.

The input data variables to the CFNN are the 60-second time-integrated values of simulated sensor signals. Therefore, most of the measured sensor noises except for biased measurement errors will be removed. To consider the biased measurement uncertainties in the measured signals that are used as inputs to the CFNN, 1% and 2% over- or under-measured input signals are assumed. Table 4 shows the degradation of the performance of the CFNN models due to the biased measurement uncertainties for the test dataset. Even though the CFNN model underperforms due to the biased measurement uncertainties, the RMS errors of the CFNN model do not exceed about 7% if the biased measurement uncertainties are maintained under 2%. Also, the RMS and maximum errors increases linearly according to the biased measurement uncertainties.

4. Conclusion

When accidents, such as LOCAs, occur in NPPs, it is important for the plant's operators to know the state of the accident quickly in order to manage it efficiently. The CFNN model presented in this paper was designed to estimate the LOCA break size using the short-time scale integrated values of five simulated sensor signals after a reactor scram. The CFNN model was developed and verified using independent development and test data sets. The performance results of the CFNN model show that the RMS error decreases as the stage number of the CFNN model increases. In addition, the performance results of the CFNN model produce an RMS error level below 0.7%. Therefore, it is confirmed that the CFNN model can accurately predict the LOCA break size. If the operators can be informed of the break size of the LOCA, it is expected that they can respond quickly and properly to the LOCA circumstances to prevent the meltdown of the reactor core.

Conflicts of interest

All authors have no conflicts of interest to declare.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant, funded by the Korean Government (MSIP) (Grant No. 2012M2B2B1055611) and the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant, funded by the Korean Government (MOTIE) (Grant No. 20164030201290).

REFERENCES

[1] M.G. Na, S.H. Shin, D.W. Jung, S.P. Kim, J.H. Jeong, B.C. Lee, Estimation of break location and size for loss of coolant accidents using neural networks, Nucl. Eng. Des. 232 (2004) 289–300.
[2] J. Garvey, D. Garvey, R. Seibert, J.W. Hines, Validation of on-line monitoring techniques to nuclear plant data, Nucl. Eng. Technol. 39 (2007) 149–158.
[3] J.W. Hines, D.J. Wrest, R.E. Uhrig, Signal validation using an adaptive neural fuzzy inference system, Nucl. Technol. 119 (1997) 181–193.
[4] M.G. Na, A neuro-fuzzy inference system for sensor failure detection using wavelet denoising, PCA and SPR, J. Korean Nucl. Soc. 33 (2001) 483–497.
[5] E.B. Bartlett, R.E. Uhrig, Nuclear power plant diagnostics using an artificial neural network, Nucl. Technol. 97 (1992) 272–281.
[6] A. Gofuku, H. Yoshikawa, S. Hayashi, K. Shimizu, J. Wakabayashi, Diagnostic techniques of a small-break loss-of-coolant accident at a pressurized water reactor plant, Nucl. Technol. 81 (1988) 313–332.
[7] M. Marseguerra, E. Zio, Fault diagnosis via neural networks: the Boltzmann machine,, Nucl. Sci. Eng. 117 (1994) 194–200.
[8] Y.G. No, J.H. Kim, D.H. Lim, K.I. Ahn, Monitoring severe accidents using AI techniques, Nucl. Eng. Technol. 44 (2012) 393–404.
[9] M.G. Na, H.Y. Yang, D.H. Lim, A soft-sensing model for feedwater flow rate using fuzzy support vector regression, Nucl. Eng. Technol 40 (2008) 69–76.
[10] S.H. Park, D.S. Kim, J.H. Kim, M.G. Na, Prediction of the reactor vessel water level using fuzzy neural networks in severe accident circumstances of NPPs, Nucl. Eng. Technol. 46 (2014) 373–380.
[11] S.H. Park, J.H. Kim, K.H. Yoo, M.G. Na, Smart sensing of the RPV water level in NPP severe accidents using a GMDH algorithm, IEEE Trans. Nucl. Sci. 61 (2014) 931–938.
[12] 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.
[13] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man. Mach. Stud. 7 (1975) 1–13.
[14] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man. Cybern. 1 (1985) 116–132.
[15] J.C. Duan, F.L. Chung, Cascaded fuzzy neural network model based on syllogistic fuzzy reasoning, IEEE Trans. Fuzzy Syst. 9 (2001) 293–306.
[16] D.Y. Kim, K.H. Yoo, G.P. Choi, J.H. Back, M.G. Na, Reactor vessel water level estimation during severe accidents using

casecaded fuzzy neural networks, Nucl. Eng. Technol. 48 (2016) 702–710.
[17] I. Lindholm, E. Pekkarinen, H. Sjovall, Evaluation of reflooding effects on an overheated boiling water reactor core in a small steam-line break accident using MAAP, MELCOR, and SCDAP/RELAP5 computer codes, Nucl. Technol. 112 (1995) 42–57.
[18] C. Allison, Comparison between MAAP, MELCOR and SCDAP/RELAP5, in: Proceedings of the Workshop on Severe Accident Research, Japan (SARJ-97), 1998, pp. 396–401.
[19] M.G. Na, W.S. Park, D.H. Lim, Detection and diagnostics of loss of coolant accidents using support vector machines, IEEE Trans. Nucl. Sci. 55 (2008) 628–636.