Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 7-8, 2015

Estimation of Minimum DNBR Using Cascaded Fuzzy Neural Networks

Dong Yeong Kim, Kwae Hwan Yoo, Man Gyun Na*
Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, Korea
**Corresponding author: magyna@chosun.ac.kr*

1. Introduction

In order to operate the pressurized water reactor (PWR) safely, the temperature of the nuclear fuel surface has to increase in the nucleate boiling region. However, a point may be reached where the bubble density becomes so great that adjacent bubbles coalesce and begin to form a vapor film across the surface of the rods. This phenomenon of boiling crisis is called a departure from nucleate boiling (DNB). The DNB phenomena can influence the fuel cladding and fuel pellets [1]. The DNB ratio (DNBR) is defined as the ratio of the expected DNB heat flux to the actual fuel rod heat flux. Since it is very important to monitor and predict the minimum DNBR in a reactor core to prevent the boiling crisis and clad melting, a number of researches have been conducted to predict DNBR values [2-10].

The aim of this study is to estimate the minimum DNBR in a reactor core using the measured signals of the reactor coolant system (RCS) by applying cascaded fuzzy neural networks (CFNN) according to operating conditions. Reactor core monitoring and protection systems require minimum DNBR prediction. The CFNN can be used to optimize the minimum DNBR value through the process of adding fuzzy neural networks (FNN) repeatedly.

The used output data of the CFNN are minimum DNBR values in a reactor core in a number of operating conditions and the input data are reactor power, core inlet temperature, pressurizer pressure, coolant flowrate of a reactor core, axial shape index (ASI), and a variety of control rod positions.

The proposed DNB estimation algorithm was verified by applying the nuclear and thermal data acquired from many numerical simulations of the optimized power reactor 1000 (OPR1000).

2. Cascaded fuzzy neural networks

The fuzzy system has been produced based on “Learning” and “Inference” intelligently. The study of fuzzy theory has been aimed in order to prove by mathematical approach about inaccuracy in thought and action of human.

2.1. Fuzzy inference system

The fuzzy inference system (FIS) generally uses the conditional rules that is comprised of *if-then* rules of the antecedent part and consequent part, and is one of the methods of artificial intelligence [11]. Both the antecedent and consequent parts have membership functions. In most cases, the Gaussian, triangular, trapezoid and bell-shaped functions are used in the formula of the membership function.

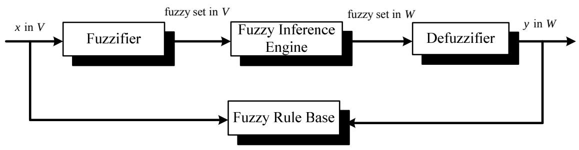

Fig. 1. Fuzzy Inference System (Mamdani-type FIS)

Fig. 1 shows the FIS. This study uses the Takagi-Sugeno-type FIS that does not need the defuzzifier in the output terminal because its output is a real value. Using the Takagi-Sugeno-type, an arbitrary i -th rule can be expressed as follows [12]:

$$\begin{aligned} &If\ x_1(k)\ is\ A_{i1}(k)\ AND \cdots AND\ x_m(k)\ is\ A_{im}(k), \\ &then\ \hat{y}^i(k)\ is\ f^i(x_1(k),\cdots,x_m(k)) \end{aligned} \tag{1}$$

where $x_j(k)$ is the input linguistic variable to the fuzzy inference model ($j=1,2,\ldots,m$; m is the number of input variables), $A_{ij}(k)$ is the membership function of the j -th input variable for the i -th fuzzy rule ($i=1,2,\ldots,n$; n is the number of rules), and $\hat{y}^i(k)$ is the output of the i -th fuzzy rule. The number of N input and output training data of the fuzzy model $z^T(k)=\left(\mathbf{x}^T(k), y(k)\right)$ (where $\mathbf{x}^T(k)=(x_1(k), x_2(k), \cdots, x_m(k))$ and $k=1, 2, \cdots, N$) were assumed to be available and the data point in each dimension was normalized. And Gaussian membership function was used because Gaussian membership function reduced the number of the parameters to be optimized. The output of the FIS using the Takagi-Sugeno-type can be expressed as follow [12]:

$$\hat{y}(k)=\sum_{i=1}^{n} y_w^i(k) \tag{2}$$

where $y_w^i(k)=\overline{w}^i(k)\, f^i(\mathbf{x}(k))$.

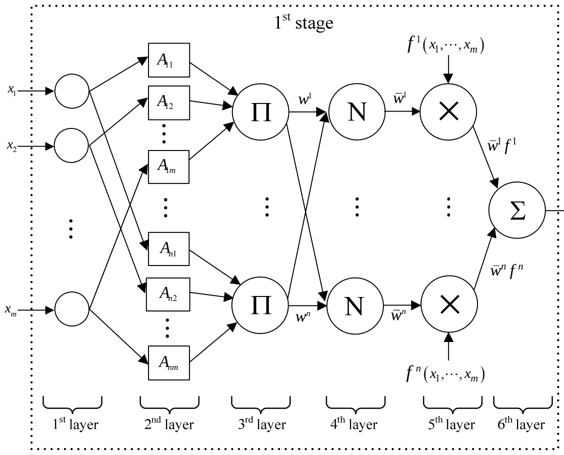

Fig. 2. Fuzzy Neural Network (FNN)

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

$$w^i(k) = \prod_{j=1}^{m} A_{ij}(x_j(k)) \tag{4}$$

$$\bar{w}^i(k) = \frac{w^i(x(k))}{\sum_{i=1}^{n} w^i(x(k))} \tag{5}$$

Fig. 2 shows the calculation method of the FIS [13]. The first layer indicates the input nodes that directly transmit the input values to the next layer. Each output from the first layer is transmitted to the input of a membership function. The second layer indicates a fuzzification layer that calculates membership function values. The third layer indicates a product operator on the membership functions that is expressed as Eq. (4). The fourth layer performs a normalization operation that is expressed as Eq. (5). The fifth layer generates the output of each fuzzy *if-then* rule. Finally, the sixth layer performs an aggregation of all the fuzzy *if-then* rules and is expressed as Eq. (2).

Therefore, the output of the FIS by Eq. (2) is expressed as the vector product as follow:

$$\hat{y}(k) = \mathbf{w}^T(k)\mathbf{q} \tag{6}$$

where

$$\mathbf{q} = \left[q_{11} \cdots q_{n1} \cdots\cdots q_{1m} \cdots q_{nm} \quad r_1 \cdots r_n\right]^T \tag{7}$$
$$\mathbf{w}(k) = \Big[\bar{w}^1(k)x_1(k) \cdots \bar{w}^n(k)x_1(k) \cdots \bar{w}^1(k)x_m(k) \cdots \bar{w}^n(k)x_m(k) \quad \bar{w}^1(k) \cdots \bar{w}^n(k)\Big]^T$$

The vector $\mathbf{q}$ is called a consequent parameter vector that has $(m+1)n$ dimensions, and the vector $\mathbf{w}(k)$ consists of input data and membership function values. The estimated output for a total of N input and output data pairs induced from Eq. (7) can be expressed as follows:

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{q} \tag{8}$$

where

$$\hat{\mathbf{y}} = \left[\hat{y}(1)\ \hat{y}(2) \cdots \hat{y}(N)\right]^T,\ \mathbf{W} = \left[\mathbf{w}(1)\ \mathbf{w}(2) \cdots \mathbf{w}(N)\right]^T$$

The matrix $\mathbf{W}$ consists of input data and membership function values. The output values of FIS are expressed in a matrix, $\mathbf{W}$, of $N \times (m+1)n$ dimensions and a parameter vector $\mathbf{q}$ of $(m+1)n$ dimensions.

2.2. Optimization of the fuzzy inference system

In this study, the CFNN is used to estimate the minimum DNBR and the training of the FNN is accomplished by a hybrid method combined with a back-propagation algorithm and a least-squares algorithm. The back-propagation algorithm that uses a gradient descent method is a general method for recursively training the fuzzy neural networks. The gradient descent method tunes the antecedent parameters (the center position of membership functions and their sharpness) so that the predefined objective function E is minimized. In order to train an antecedent parameter a_{ij}, the following iterative calculation is used:

$$a_{ij}(t+1) = a_{ij}(t) - \eta_a \frac{\partial E}{\partial a_{ij}}\Big|_t \tag{9}$$

where

$$E = \sum_{k=1}^{N}\left(y_k - \hat{y}_k\right)^2,$$
$$i = 1, 2, \cdots, n$$
$$j = 1, 2, \cdots, m$$
$$t = 0, 1, 2, \cdots$$

η_a is a learning rate for a parameter a. The gradient descent method is very stable when the learning rate is small but susceptible to local minimum.

If antecedent parameters of the FIS are determined by the back-propagation algorithm, the resulting fuzzy neural networks are equivalent to a series of expansions of some basis functions. This basis function expansion is linear in its adjustable parameters. Therefore, the least-square method was used to determine the consequent parameter of fuzzy rules. The consequent parameter $\mathbf{q}$ was chosen to minimize the following objective function. This objective function consists of the square error between the actual value y and its predicted value $\hat{y}$, and it is expressed as follows:

$$J = \sum_{k=1}^{N_t} \left[y(k) - \hat{y}(k) \right]^2 = \sum_{k=1}^{N_t} \left[y(k) - \mathbf{w}^T(k)\mathbf{q} \right]^2 \tag{10}$$

where

$$\mathbf{y} = \left[y(1) \;\; y(2) \;\; \cdots \;\; y(N_t) \right]^T$$

$$\hat{\mathbf{y}} = \left[\hat{y}(1) \;\; \hat{y}(2) \;\; \cdots \;\; \hat{y}(N_t) \right]^T$$

N_t is the number of training data.

A solution for minimizing the above objective function can be obtained using Eq. (8). To solve the parameter vector, $\mathbf{q}$, the inverse of the matrix $\mathbf{W}$ must exists. On the other hand, there is no inverse matrix generally. Therefore, the pseudo-inverse of the matrix $\mathbf{W}$ was used. The parameter vector, $\mathbf{q}$, is solved easily from the pseudo-inverse as shown below.

$$\mathbf{q} = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T \mathbf{y} \tag{11}$$

That is, the parameter vector $\mathbf{q}$ can be calculated from a series of input and output data pairs.

2.3. Cascaded fuzzy neural networks

There have been a number of studies on the fusion of fuzzy logic and neural networks, termed FNN. Most of the existing FNN models have been proposed to implement different types of single-stage fuzzy reasoning mechanisms. However, single-stage fuzzy reasoning is only the most simple among a human being's various types of reasoning mechanisms. Syllogistic fuzzy reasoning, where the consequence of a rule in one reasoning stage is passed to the next stage as a fact, is essential to effectively build up a large scale system with high level intelligence [14]. In view of the fact that the fusion of syllogistic fuzzy logic and neural networks has not been sufficiently applied in nuclear engineering field, a CFNN model based on syllogistic fuzzy reasoning is applied in this paper.

The CFNN model contains two or more inference stages where each stage corresponds to a single-stage FNN module. Each single-stage FNN module contains fuzzification, fuzzy inference, and training units. The architecture of the CFNN is shown in Fig. 3.

The CFNN can be used to estimate the target value through the process of adding FNN repeatedly. In CFNN method, the first stage FNN is the same as the FNN of Fig. 2. The second stage FNN uses the initial input variables and the output variable of the first stage FNN as input variable. Therefore, this process is repeated L times to find the optimum value.

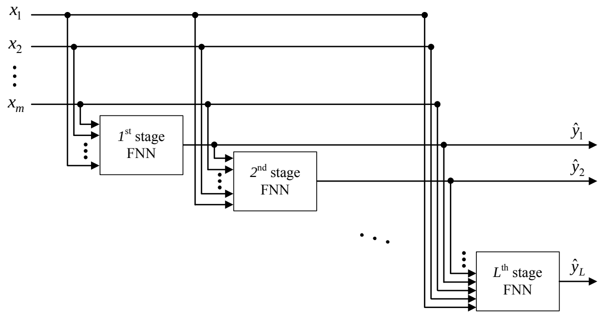

Fig. 3. Cascaded Fuzzy Neural Network (CFNN)

Similarly to Eq. (1), an arbitrary i-th rule of the CFNN can be expressed as Eq. (12):

$$\begin{aligned}
&\text{Stage 1} \begin{bmatrix} \textit{If } x_1(k) \textit{ is } A_{i1}^1(k) \textit{ AND} \cdots \textit{AND } x_m(k) \textit{ is } A_{im}^1(k), \\ \textit{then } \hat{y}_1^i(k) \textit{ is } f_1^i(x_1(k), \cdots, x_m(k)) \end{bmatrix} \\
&\text{Stage 2} \begin{bmatrix} \textit{If } x_1(k) \textit{ is } A_{i1}^2(k) \textit{ AND} \cdots \textit{AND } x_m(k) \textit{ is } A_{im}^2(k) \\ \textit{AND } \hat{y}_1(k) \textit{ is } A_{i(m+1)}^2(k), \\ \textit{then } \hat{y}_2^i(k) \textit{ is } f_2^i(x_1(k), \cdots, x_m(k), \hat{y}_1(k)) \end{bmatrix} \\
&\qquad \vdots \\
&\text{Stage } L \begin{bmatrix} \textit{If } x_1(k) \textit{ is } A_{i1}^L(k) \textit{ AND} \cdots \textit{AND } x_m(k) \textit{ is } A_{im}^L(k), \\ \textit{AND } \hat{y}_1(k) \textit{ is } A_{i(m+1)}^L(k) \textit{ AND} \cdots \textit{AND } \hat{y}_{(L-1)}(k) \textit{ is } A_{i(m+L-1)}^L(k), \\ \textit{then } \hat{y}_L^i(k) \textit{ is } f_L^i(x_1(k), \cdots, x_m(k), \hat{y}_1(k), \cdots, \hat{y}_{(L-1)}(k)) \end{bmatrix}
\end{aligned} \tag{12}$$

where L is the total number of stages and the remaining variables are the same as before.

The CFNN model is trained sequentially at each FNN module by the same way as explained in subsection 2.2. Fig. 4 shows the optimization procedure of the CFNN model.

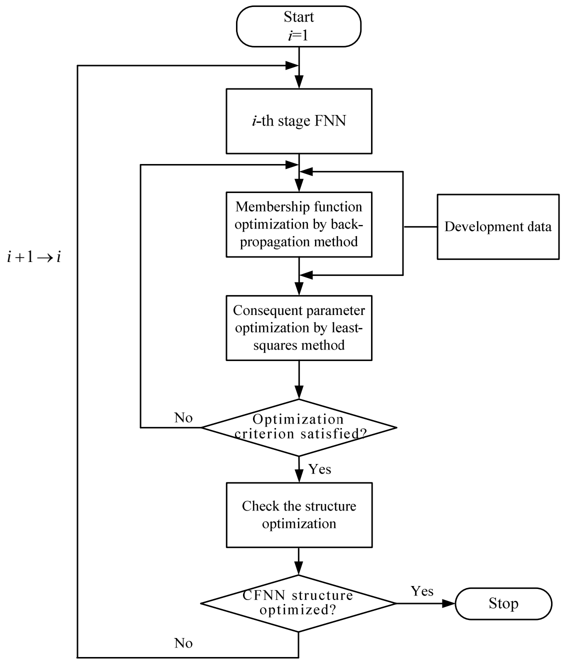

Fig. 4. Optimization procedure of CFNN model

3. Application to the minimum DNBR estimation

The proposed algorithm was applied to the first fuel cycle of the OPR1000. The DNB data were obtained by running the MASTER [15] and COBRA codes [16]. The MASTER (Multipurpose Analyzer for Static and Transient Effects of Reactor) reactor analysis code developed by KAERI (Korea Atomic Energy Research Institute) is a nuclear analysis and design code which can simulate the PWR and BWR core in 1-, 2-, 3-dimensional geometry. The MASTER code was designed to have a variety of capabilities such as static core design, transient core analysis and operation support and is interfaced with the COBRA code for thermo-hydraulic calculations. Since these two codes are best-estimated codes, additional margins should be provided to setup the protection limits for DNB protection or alarm set points for DNB monitoring [10].

The DNB data comprise a total of 18816 input-output data pairs ($x_1, x_2, \cdots, x_9, y_r$) that can describe the reactor core states appropriately in the ranges of the input variables given in Table I [10]. In this study, the used DNB data were composed of 200 pieces of test data and the remaining data were used to development the CFNN model, which is a development data set. 90% in the development data set was used to train each FNN module and 10% was used to optimize the CFNN structure. x_1 through x_9 are the input signals that represent the reactor power, core inlet temperature, coolant pressure, mass flowrate, axial shape index (ASI), R2, R3, R4 and R5 control rod positions, and y_r is the output signal which indicates the minimum DNBR in the reactor core. ASI is defined as $(P_B - P_T)/(P_B + P_T)$ where P_B is the bottom-half power and P_T is the top-half power of a nuclear reactor.

Table I: Ranges of input and output signals

Input signals	*Nominal values*	*Ranges*
Reactor power (%)	100%	80 ~ 103
Inlet temperature (°C)	295.8	290.5 ~ 301.7
Pressure (bar)	155.17	131.0 ~ 160.0
Mass flowrate (kg/m^2-sec)	3565.0	2994.6 ~ 4135.4
ASI	-	-0.597 ~ 0.534
SPND signals	-	7.4 ~ 322.0
R2 control rod positions (cm)	-	0 ~ 381
R3 control rod positions (cm)	-	0 ~ 381
R4 control rod positions (cm)	-	0 ~ 381
R5 control rod positions (cm)	-	0 ~ 381
Output signals	*Nominal values*	*Ranges*
DNBR value	-	0.853 ~ 5.176

The DNB data were divided into the development data and test data sets. The CFNN is trained for two DNBR development data sets divided into both the positive (relatively high at a bottom part of a reactor core) ASI and the negative ASI since these results had smaller errors compared with results with only one data set. The number of rules of the CFNN is 6.

Table II: DNBR calculation results by the CFNN

	Training data			Test data		
	No. of data points	RMS error (%)	Relative maximum error (%)	No. of data points	RMS error (%)	Relative maximum error (%)
Positive ASI	8467	0.181	0.920	100	0.229	1.152
Negative ASI	8467	0.128	0.856	100	0.116	0.289

Table II summaries the DNBR calculation results by the CFNN. If ASI value is positive, the RMS error and the relative maximum error are 0.23% and 1.15%, respectively. Also, if ASI value is negative, the RMS error and the relative maximum error are 0.12% and 0.29%, respectively.

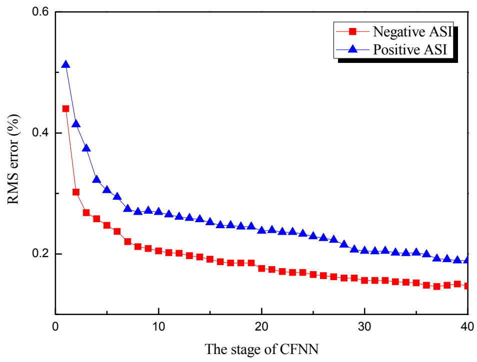

Fig. 5. RMS error versus the number of stage of the CFNN (development data)

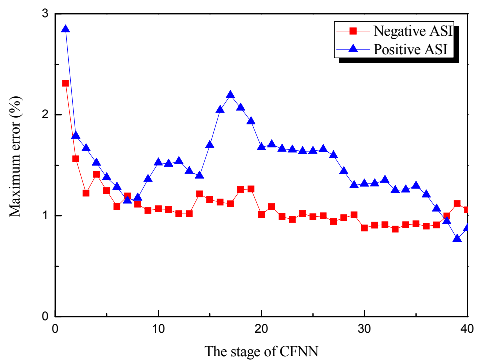

Fig. 6. Maximum error versus the number of stage of the CFNN (development data)

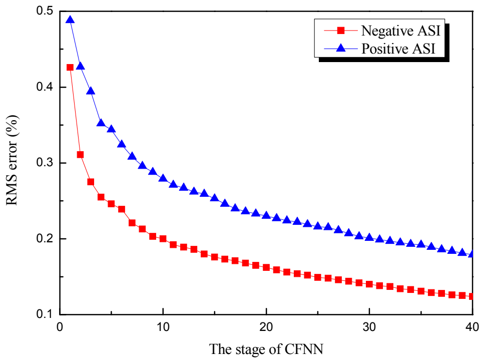

Fig. 7. RMS error versus the number of stage of the CFNN (test data)

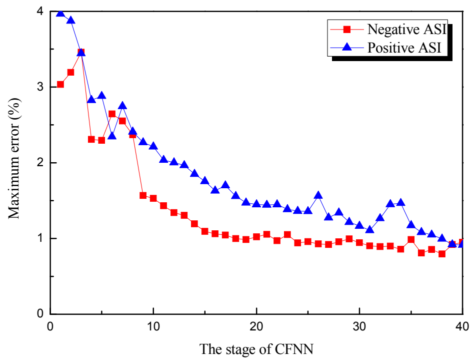

Fig. 8. Maximum error versus the number of stage of the CFNN (test data)

Fig. 5 and 6 show RMS error and maximum error for the development data, respectively. As the number of stages of the CFNN is increased, the errors are reduced gradually. Fig. 7 and 8 show RMS error and maximum error for the test data, respectively. As the number of stages of the CFNN is increased, the errors are reduced gradually. The sequential execution of the CFNN was carried out until performance change according to the number of stage of the CFNN is not large.

Table III shows the results of the FNN model [8] and the fuzzy support vector regression (FSVR) model [9] developed previously. It is shown that the proposed the CFNN model has better performance compared to the existing FNN and FSVR models.

Table III: DNBR calculation results by FNN [8] and FSVR [9] models

	FNN		FSVR	
	RMS error (%)	Relative maximum error (%)	RMS error (%)	Relative maximum error (%)
Positive ASI	0.375	10.542	0.320	1.969
Negative ASI	0.258	7.697	0.255	0.854

4. Conclusion

In this paper, CFNN models have been developed to estimate the minimum DNBR in the reactor core. The proposed algorithm is trained by using the data set prepared for training (development data) and verified by using another data set different (independent) from the development data.

The developed CFNN models were applied to the first fuel cycle of OPR1000. The RMS errors are 0.23% and 0.12% for the positive and negative ASI, respectively. The CFNN is sufficiently accurate to be used in DNBR estimation.

REFERENCES

[1] L.S. Tong, "An evaluation of the departure from nucleate boiling in bundles of reactor fuel rods," *Nucl. Sci. Eng.,* vol. 33, pp. 7-15, Jan. 1968.

[2] H.C. Kim and S.H. Chang, "Development of a back propagation network for one-step transient DNBR calculations," *Ann. Nucl. Energy*, vol. 24, no. 17, pp.1437-1446, Nov. 1997.

[3] S. Han, U.S. Kim, and P.H. Seong, "A methodology for benefit assessment of using in-core neutron detector signals in core protection calculator system (CPCS) for Korea standard nuclear power plants (KSNPP)," *Ann. Nucl. Energy*, vol. 26, no. 6, pp. 471-488, April 1999.

[4] M.G. Na, "On-line estimation of DNB protection limit via a fuzzy neural network," Nucl. Eng. Tech., vol. 30, no. 3, pp. 222-234, June 1998.

[5] M.G. Na, "DNB limit estimation using an adaptive fuzzy inference system," IEEE Trans. Nucl. Sci., vol. 47, no. 6, pp. 1948-1953, Dec. 2000.

[6] W.K. In, D.H. Hwang, Y.J. Yoo, and S.Q. Zee, "Assessment of core protection and monitoring systems for an advanced reactor SMART," Ann. Nucl. Energy, vol. 29, no. 5, pp. 609-621, March 2002.

[7] G.C. Lee, W.P. Baek, and S.H. Chang, "Improved methodology for generation of axial flux shapes in digital core protection systems," Ann. Nucl. Energy, vol. 29, no. 7, pp. 805-819, May 2002.

[8] G.C. Lee and S.H. Chang, "Radial basis function networks applied to DNBR calculation in digital core protection systems," Ann. Nucl. Energy, vol. 30, no. 15, pp. 1561-1572, Oct. 2003.

[9] M.G. Na, S.M. Lee, S.H. Shin, D.W. Jung, K.B. Lee, and Y.J. Lee, "Minimum DNBR monitoring using fuzzy neural networks," Nucl. Eng. Des., vol. 234, no. 1-3, pp. 147-155, Dec. 2004.

[10] S.W. Lee, D.S. Kim, and M.G. Na, "Prediction of DNBR Using Fuzzy Support Vector Regression and Uncertainty Analysis," IEEE Trans. Nucl. Sci., Vol. 57, No. 3, pp. 1595-1601, June 2010.

[11] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," Int. J. Man-Machine Studies, vol. 7, pp. 1-13, 1975.

[12] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," IEEE Trans. Systems, Man, Cybern., vol. SMC-15, no. 1, pp. 116-132, Jan./Feb. 1985.

[13] D.Y. Kim, K.H. Yoo, J.H. Kim, M.G. Na, S. Hur, and C-H. Kim, "Prediction of Leak Flow Rate Using Fuzzy Neural

Networks in Severe Post-LOCA Circumstances," IEEE Trans. Nucl. Sci., Vol. 61, No. 6, pp. 3644-3652, Dec. 2014.

[14] J.C.Duan and F,L Chung, "Cascaded Fuzzy Neural Network Model Based on Syllogistic Fuzzy Reasoning," IEEE Trans. Fuzzy Systems, vol. 9, no. 2, pp.293-306, Apr. 2001.

[15] B.O. Cho, H.G. Joo, J.Y. Cho, and S.Q. Zee., "MASTER: reactor core design and analysis code," in *Proc. 2002 Int. Conf. New Frontiers of Nuclear Technology: Reactor Physics (PHYSOR 2002)*, Seoul, Korea, Oct. 7-10, 2002.

[16] C.L. Wheeler, C.W. Stewart, R.J. Cena, D.S. Rowe, and A.M. Sutey, "COBRA-IV-I; An interim version of COBRA for thermal hydraulic analysis of rod bundle nuclear fuel elements and cores," BNWL-1962, March 1976.