Proceedings of the 14th International Conference on Nuclear Engineering
ASME 2011
September 28-30, 2011, Washington DC, USA

SMR2011-6544

COMPUTATIONAL INTELLIGENCE AS A TOOL FOR SMALL MODULAR REACTORS

Milos Manic
University of Idaho
Idaho Falls, ID, USA

Piyush Sabharwall
Idaho National Laboratory
Idaho Falls, ID, USA

ABSTRACT

Computational intelligence techniques (CITs) traditionally consist of artificial neural networks (ANNs), fuzzy systems and genetic algorithms. This article overviews diverse implementations of ANNs, which are the most prominent in nuclear engineering problems, especially for small modular reactors (SMRs). Advanced computational intelligence-based tools will allow data to be transformation into knowledge, thus improving understanding, predictability (can be seen from the two case studies for thermal-hydraulic prediction), sustainability, and performance of SMRs with real time analysis and monitoring.

I INTRODUCTION

Next generation reactor designs of small modular reactors (SMRs) include light water reactors, gas cooled reactors, liquid metal fast reactors, and molten salt reactors. Potential major markets for small to medium sized reactors are in areas where there is (1) a demand for energy in the form of electricity or process heat, (2) no access to a main power grid, and (3) not enough power required to make a larger reactor economical. Compared with larger reactors, small to medium sized reactors are safer, more economical, easier to operate, can be modularly constructed, provide a faster return on investment, and have a lower capital investment, thereby encouraging a variety of potential users. Irrespective of the design type, SMRs will have nonlinearities and challenges similar to large reactors, maybe even more. Thus, a tool that would provide and enhance predictive capabilities, such as ANNs, could further improve user understanding of the physical processes.

ANNs are known as universal approximators and classifiers because they have unique capabilities in dealing with high-dimension, highly-nonlinear data. ANNs have several advantageous features. When the number of modeled inputs of a system increases, the ANN architectural solution remains the same, i.e. the computational complexity of the system, memory storage requirements, and computational time does not increase. ANNs are data-driven and can "mimic" the system behavior by adaptively learning input/output interrelationships among system variables through gradient search. By capturing nonlinear interrelationships. ANNs exhibit highly accurate predictive capabilities that can improve understanding of physical processes. These advantages make ANNs a valuable and viable tool for predicting, optimizing, and controlling SMR design problems.

As powerful, universal, multidimensional approximators, ANNs are a valuable and viable tool in nuclear engineering and design problems that may include optimization, prediction, classification, and nonlinear control. Examples of predictive features of ANNs include thermal-hydraulic performance analysis of the printed circuit heat exchangers (PCHE) [Kim 09, Ridluan 2009], and prediction of key safety parameters in nuclear research reactors [Mazrou 2009]. ANNs have been used to optimally design compact heat exchangers (CHE) [Jia 2003] and to optimize plate elements or plate-fin heat exchangers (PFHE) of reactors [Waheda 2010, Peng 2007]. Control modeling and simulation related ANN applications include feedwater controllers in pressurized water reactors (PWRs) [Jia 2003].

Computational intelligence techniques (CITs) will employ state-of-the-art data mining techniques built upon established paradigms of pattern recognition, high performance computing, and data visualization. The data-mining component is focused on discovering knowledge and extracting significant historical patterns. The extracted significant patterns and mined root-cause information will provide vital clues to understanding and detecting symptoms before instrumentation and control systems fail.

Section II of this paper gives a high-level introduction to artificial neural networks and Section III overviews applications of computational intelligence, followed by two case studies.

II ARTIFICIAL NEURAL NETWORKS

This section presents a high level introduction to artificial neural networks.

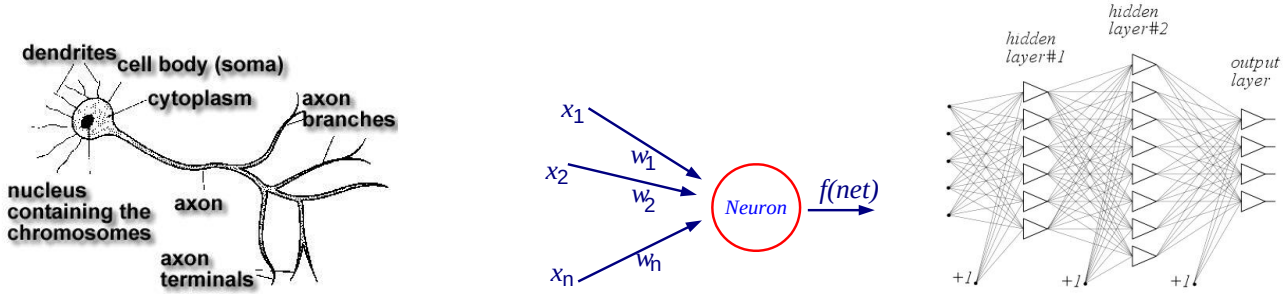
A. NEURAL NETWORKS, BASICS

A single biological neuron (Fig. 1a) connects to other neurons via its interconnections (synapses), and receives the stimulated input via its inputs (dendrites). A weighted sum of input signals (net value) is then compared against the threshold or certain activation function. An output signal produced in such a way is further transmitted via other synapses to other neurons. Artificial neurons are biologically inspired computational structures that can be viewed as summation threshold elements (Fig. 1b). ANNs can be composed of several layers of neurons, fully or partially connected, with optional feedback connections (Fig. 1c). Such ANNs represent computational architecture credited with powerful features such as universal approximators and classifiers.

Artificial neurons represent summation-threshold elements (Fig. 2). Weighted sum of the neuron inputs (x_i) is referred to as *net*. When *net* exceeds predefined threshold value, the neuron produces an output, i.e., "fires" (1):

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}, \quad out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases} \qquad (1)$$

The weight (w_{i+1}) with default input +1 is called *bias*, and can be understood as the threshold (T), but with the opposite sign (Fig. 2c).



a) *b)* *c)*

Fig 1. a) Biological neuron; b) artificial counterpart of biological neuron; c) artificial neural network

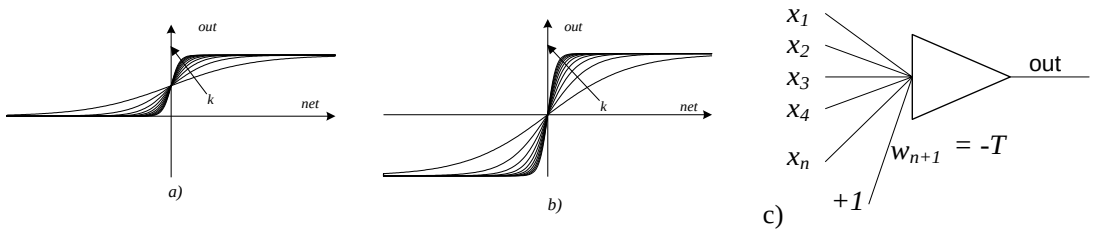


Fig 2. a) Typically used threshold functions: unipolar and bipolar sigmoidal; b) Artificial neuron as weighted threshold element.

The output is calculated via so called activation (transfer or threshold) function. Typically, used threshold functions are sinusoidal threshold functions, both unipolar and bipolar (Fig. 2a). They exhibit behavior typical in biological systems in that they respond to the sensory inputs until saturation (observed for pain or taste for example; maximum is reached at some point). In other words, for any combination of inputs (net value), activation function constrains output to values between 0 and 1 (unipolar), and -1 and +1 (bipolar). In this way, neurons exhibit the range of net values for which learning occurs, and range of net values where saturation occurs (incremental changes of large +/- values of net do not yield response change of a neuron). The typically used bipolar sinusoidal activation functions can be defined as (Fig. 2b)

$$o_{bip} = f_{bip}(k \times net) = \tanh(k \times net) = \frac{2}{1 + exp(-2 \times k \times net)} - 1 \qquad (2)$$

The graphical representation of a single neuron operation can be described easily via analytic geometry. For example, a simple two-dimensional (2-D) neuron represents a separation line (Fig. 3), so a single, two input neuron represents a linear classifier where the neuron definition is $\times + 3y - 3 > 0$, where values 1 and 3 are the weights used for inputs x and y. The neuron divides the xOy space into two areas, by selecting the upper one. In this example, the neuron correctly classifies the rectangular pattern producing the output of +1, identical to the desired output of +1. Further, by deselecting the lower part of xOy space, the neuron produces -1 on the output, again matching the desired output of -1. Similarly, a 3-D neuron represents a separation plane, while neurons with n dimensions

represent a hyperplane in *n*-dimensional space R^n. It is already obvious that a single neuron can be a very powerful computational element.

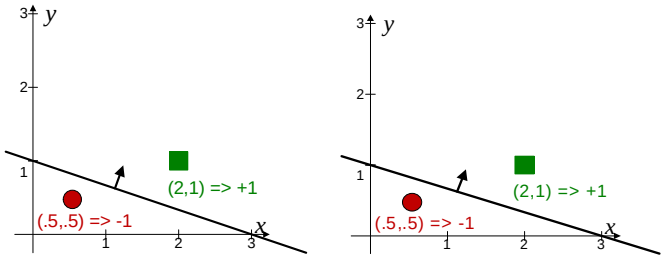


Fig. 3. Graphical representation of single neuron operation.

One of the most used ANN algorithms is Error Back Propagation (EBP), proposed by Werbos [1994] and Rumelhart [1986]. Other popular algorithms include modification of EBP (Quickprop, RPROP, Delta-Bar-Delta, Back Percolation), Levenberg-Marquardt (LM), Adaptive Resonance Theory, Counter Propagation Networks, and Cascade Correlation Networks.

B. TWO WIDELY USED ALGORITHMS (EBP, LM)

Error Back Propagation Algorithm

The fundamental principles of neural networks design is explained by EBP, one of the most frequently used approaches.

The EBP represents one of the seminal algorithms in neural networks design research. Although not without concerns when it comes to robustness to parameter initialization and convergence, it was historically the first algorithm to introduce multiple layer neural networks activated by sigmoid (*tanh)* transfer function. Hence, it enabled the solving of linearly nonseparable problems.

Introduced by Werbos [1974, 1994] and later by Rumelhart and McClelland [1986], EBP represented a real breakthrough in the ANN by offering a modern algorithm capable of solving a plethora of problems of universal functional learning and clustering and other complex and highly nonlinear multidimensional problems.

As the original name of the algorithm implies, EBP is composed of two phases: the forward phase (when inputs are propagated through the network), and the backward phase (where errors between expected and actual outputs of the network are propagated back through the network). During the backward propagation of errors, weights of neurons that produced the error are modified accordingly, going backwards from one layer to another, as illustrated in Fig. 4 [Manic (i) 2011].

The weight set of a neural network is typically updated as:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta\mathbf{w} \qquad (3)$$

where weight matrix in current, $k+1^{th}$ iteration, w_{k+1}, is calculated based on the matrices from previous k^{th} iteration, weight matrix w_k and weight increment $\Delta\mathbf{w}$.

In order to minimize the output error, the weight increment in equation (3) is calculated against the gradient change as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha\,\mathbf{g}, \text{ where } \Delta w_i = -\alpha \cdot gradient \qquad (4)$$

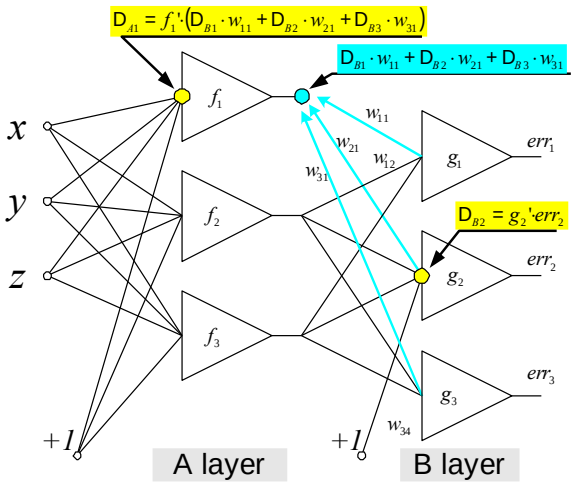


Fig. 4. Error-Back Propagation algorithm, forward and backward phase.

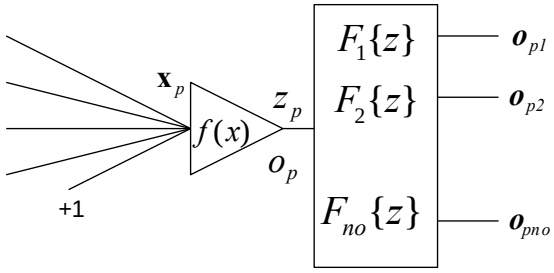


Fig. 5. Error-Back Propagation algorithm—individual neuron vs. output layer schematics.

Equation (4) is also known as the Steepest Descent Method. The gradient search typically expresses search for a weight set that would minimize the total error defined as

$$TE = \sum_{p=1}^{np} \left(d_p - o_p\right)^2 \qquad (5)$$

where output (o_p), i.e., output from the *pth* neuron of output layer is defined as function composition—function *F (*nonlinear signals propagation through layers as shown in Fig. 5) of function *f* (activation function) of net

$$o_p = F\left\{ f\left(w_1 x_{p1} + w_2 x_{p2} + \cdots + w_n x_{pn}\right)\right\}. \qquad (6)$$

Hence, the gradient becomes:

$$gradient = \frac{d\left(TE\right)}{dw_i} = -\sum_{p=1}^{np} \left[2\left(d_p - o_p\right)\frac{do_p}{dz_p}\frac{dz_p}{dnet_p}\frac{dnet_p}{dw_i}\right] \qquad (7)$$

and the weight increment $\Delta\mathbf{w}$ becomes:

$$\Delta \mathbf{w}_p = \alpha \cdot \sum_{o=1}^{no} \sum_{p=1}^{np} \left[(d_{op} - o_{op}) \cdot F'\{z_p\} \cdot f'(net_p) \cdot \mathbf{x}_p \right] \quad (8)$$

where x_p represents network input pattern, while d_{op}, z_p, and o_{op} are desired, neuron, and network outputs, respectively. Here $f'(net_p)$ is the derivative of activation function f with respect to the neuron net value (net_p), while $F'\{z_p\}$ is the derivative of nonlinear function $F\{z_p\}$ describing the propagation of nonlinear mapping through the multiple network layers, and (z_p) is the individual neuron output (other than output layer neuron).

Levenberg-Marquardt Algorithm

LM is another frequently used algorithm [Levenberg 1994; Marquardt 1994]. The process is similar to the Equations (1)–(5) of EBP approach, except that (4) becomes Newton's method:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{A}_k^{-1}\mathbf{g}, \; \Delta\mathbf{w}_k = -\mathbf{A}_k^{-1}\mathbf{g}, \quad (7)$$

i.e., process is accelerated by replacing learning constant alpha with the Hessian, the 2nd order derivative matrix.

After applying the following substitutions to (7),

$$\mathbf{A} \approx 2\mathbf{J}^T\mathbf{J}, \; \mathbf{g} = 2\mathbf{J}^T\mathbf{e} \quad (8)$$

the steepest descent becomes the LM algorithm

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (J_k^T J_k + \mu I)^{-1} J_k^T e. \quad (9)$$

The LM algorithm combines the speed of the Newton algorithm with the stability of the steepest descent method. It is important to note that the parameter μ "drives" the LM algorithm. For $\mu=0$, the LM algorithm becomes the Gauss-Newton method, while for very large μ the LM algorithm is reduced to the steepest decent [Manic (i) 2011].

III APPLICATIONS OF COMPUTATIONAL INTELLIGENCE TECHNIQUES IN NUCLEAR ENGINEERING AND DESIGN

This section briefly overviews recent advances in CITs in nuclear engineering and design.

A. NEURAL NETWORKS IN THERMAL-HUDRAULIC PREDICTION (BRIEF OVERVIEW)

The U.S. Department of Energy is leading a number of initiatives such as the Next Generation Nuclear Plant [Schultz 2004], also known as the very high temperature (gas-cooled) reactor. The key to overall plant efficiency for this initiative, like any other higher temperature energy system, is the power conversion system based on an efficient heat exchanger. The typical tradeoff that occurs in design requirements of heat exchangers are compactness (to minimize material costs) versus thermal efficiency, resulting in nonlinear and multidimensional exercises in parametric design optimization. The traditional design engineering in nuclear industry has been relying upon trial-and-error iterative methods with design constraints typically conservatively observing margins of safety in operation and off-normal anticipated and unanticipated scenarios.

The multidimensional and nonlinear behaviors observed during the process enhance the interest in ANNs as a universal, powerful, and multidimensional approximator that can advance these practices in terms of efficient system performance design and optimization, all the while meeting regulatory (licensing) compliance [Ridluan 2009].

The ANN approach has been considered in limited nuclear science and technology literature. For example, Eryurek and Upadhyaya studied the possibility of employing neural networks to model signals from a commercial power plant and the Experimental Breeder Reactor-II (EBR-II) at Idaho National Laboratory [Eryurek 1990]. Roh et al. [1991] developed a system of thermal power prediction in nuclear power plant by combining a neural network with a signal validation model. Further, Guo and Uhrig applied a hybrid type of neural networks to predict the heat rate, as linked to nuclear power plant performance [Guo 1992].

Boroushaki et al. [2005] applied cellular neural network to simulate reactor core kinetics. Guanghui et al. trained artificial neural networks to predict the critical heat flux under low pressure and oscillating conditions for both natural and forced circulation [Guanghui 2003]. Garg et al. [2007] applied multilayer perceptron and radial basis function neural networks to predict thermal-hydraulics of natural circulation boiling-water reactor. Finally, Vaziri et al. [2007] applied radial basis function and multilayer perceptron neural networks to also predict the critical heat flux. ANN is evidently receiving consideration as a suitable in-reactor analysis tool.

In the last decade, artificial neural networks have been introduced to evaluate, design, and optimize thermal-hydraulics performance of compact heat exchangers. Diaz and Sen [1999] developed four layers of artificial neural networks with sigmoid activation function to predict the heat transfer rates for 1-D conduction, 1-D convection with one and two heat transfer coefficients, and single-row PFHE. The algorithm used EBP to yield maximum error less than 3.7%. Pacheco-Vega [2001] applied feed forward structure with sigmoid function to fin-plate type heat exchanger analysis for refrigeration application. Again, the algorithm of choice was EBP, and the root-mean square error estimation indicated predictions by ANN to be at the level of the uncertainty of experiment (the details on experimental correlation and the ANN model for steady-state performance of plat fin-tube heat exchanger can be found in Pacheco-Vega's dissertation [2002]. Recently, Ping and Ling proposed the combination of GA and BP ANN to optimize the PFHE size and capital cost [Peng 2007]. Ermis [2007] applied ANN to estimate heat transfer coefficient, pressure drop, and Nusselt number. Here, the 15-channel-configuration compact heat exchangers with staggered cylindrical and triangular rib were chosen for testing the neural net approach. The algorithm of choice was the a Feed-Forward Back-Propagation algorithm.

These documented ANN-based heat exchanger analyses are targeting extended surface-type, traditional heat exchangers with heat transfer media at conventional thermophysical conditions. Only very recent literature addresses compact heat exchangers with microchannels using supercritical fluids.

Ridluan et Al. [2009] demonstrated the predictive ability of an ANN-based approach to assess PCHE thermal hydraulics with the heat transfer medium near or at critical conditions. The application focused on supercritical CO_2, which has been identified as a suitable fluid for both convective heat transfer through a single tube and a heat exchanger with multiple zigzag microchannels (PCHE).

B. NEURAL NETWORKS IN THERMAL-HYDRAULIC PREDICTION, CASE STUDY

This section summarizes two case studies of ANN design and optimization applied to thermal-hydraulic prediction based on work by Ridluan et al. [2009] and Wijayasekara et al. [2011].

Case Study 1

The experimental datasets used in this study were based on work by He [2005] and Ishizuka [2005]. He's group conducted experiments and simulations of convective heat transfer in a vertical microchannel with SCO_2 as the heat transfer medium. They validated the simulations with experimental data (excerpt provided in Table 1). He's experiments were carried out in a stainless steel (1CR189NT) vertical tube with internal and external diameters of 0.948 and 1.729 mm respectively. The critical pressure range of SCO_2, is 8.5–9.5 MPa. The test section, 55 mm in length, was heated by passing a low voltage alternating current. The experimental uncertainty was reported to be 11.3%. In this case study, He's work considered the following parameters: inlet pressure (P_i), inlet temperature (T_i), mass flow rate ($\dot{m}$),Reynolds number (Re), and Buoyancy parameter (Bo).

The second dataset used in this study was taken from Ishizuka et al. [2005]; an excerpt is presented in Table 2. The PCHE load capacity in Ishizuka's experiment was 3 kw, and each PCHE plate consists of 12 hot and 11 cold zigzagged flow channels, while the configuration of PCHE path was semicircular and zigzagged. The hot (P_{hi}) and cold (P_{ci},) inlet pressure, hot (T_{hi}) and cold (T_{ci}) temperature, and mass flow rate (G) were considered as input factors that influence three outputs: cold-sided pressure drop (DP_c), hot-sided pressure drop (DP_h), and heat transfer (Q).

Table 1. Excerpt from thermal-hydraulic database by He et al. [2005]

Mass Flow Rate $\dot{m}$ (kg/h)	Inlet Temperature T_i (°C)	Inlet Pressure P_i (MPa)	Reynolds Number Re	Buoyancy Number Bo	Wall Heat Flux (kW/m^2)
1.48	32.7	9.59	9,237	936×10^{-10}	31,534
1.53	37.8	9.54	11,639	790×10^{-10}	31,194
1.49	39.6	9.5	12,629	796×10^{-10}	30,722
1.37	51	9.43	20,864	208×10^{-10}	29,400
1.49	44	8.48	24,138	174×10^{-10}	37,079

Table 2. Excerpt from thermal-hydraulic database by Ishizuka et al. [Ishizuka 2005]

Mass Flow Rate $\dot{m}$ (kg/h)	Inlet Hot CO_2 Pressure $P_{h,i}$ (MPa)	Inlet Cold CO_2 Pressure $P_{c,i}$ (MPa)	Inlet Hot CO_2 Temperature $T_{h,i}$ (°C)	Inlet Cold CO_2 Temperature $T_{c,i}$ (°C)	Cold CO_2-sided Pressure Drop ΔP_c (kPa)	Hot CO_2-sided Pressure Drop ΔP_h (kPa)	Heat Transfer Q (kW)
42.8	2.26	6.59	280.1	107.8	34.93	9.96	2.067
52.6	2.22	6.53	280.2	107.8	53.52	15.23	2.539
79.6	2.5	7.34	279.9	107.9	93.07	26.66	3.860
52.1	3.34	10.08	280.1	108.2	32.83	10.13	2.601

Convective Heat Transfer CO_2 through a Single Tube

The first study refers to the work of Ridluan et. al. applied to a heat exchanger using supercritical CO_2 as the heat transfer medium on the convective heat transfer in a (straight) mini-tube [Ridluan 2009]. The presented EBaLM-THP algorithm was based on the specific neural network algorithm, which combined the best of the EBP and LM algorithms. The EBaLM-THP combined that robustness to the parameter initialization of EBP and the speed of LM into one algorithm.

In the first example, EBaLM-THP algorithm was tested on the problem of supercritical CO_2 flow through a straight tube. One half of the data was used for training, the other half for testing. Training and testing datasets are illustrated in terms of

heat flux (q''_w) versus temperature (T) in Fig. 6. All of values are normalized by the corresponding maximum values. That is, $T_{i,max}$ = 51°C and $q''_{w,\max}$ = 200,000 kw/m^2, both as noted by He et al. [2005]. As shown, both the training and testing datasets fluctuate but are closely matched and bounded.

The *TE* was calculated for various three-layered NN architectures (architectures with other number of layers demonstrated much higher error). A graphical illustration of *TE* versus select neural network architectures is presented in Fig. 7. For the architectures ANN 7-n-1 and ANN 8-n-1, the *TE* versus the total number of neurons is nearly identical.

Fig. 8 depicts the apparent effectiveness of ANN 6-6-1, ANN 7-5-1, and ANN 8-4-1 architectures. In spite of the highly fluctuating data, these networks were able to successfully learn the behavior of the heat exchanger with supercritical CO_2.

Convective Heat Transfer through PCHE

For convective heat transfer through PCHE, the training and testing data for the PCHE were plotted against mass flow rate (G) as shown in Fig. 9 (please refer to Ridluan [2009] for hot-sided pressure (P_h), and cold-sided pressure (P_c)). All of the values were normalized by the corresponding maximum values, as follows: ΔP_{cmax}=93.07 kPa, G_{max}=87 kg/h, ΔP_{hmax}= 26.66 kPa, $P_{hmax,i}$=3.34 MPa, and Q_{max}=4.324 kW.

Comparison of ANN versus Polynomial Fitting (FT)

To demonstrate the predictive ability of EBaLM-THP, the ANN approach was also compared against a 10th order polynomial 'fit' (Matlab) of the reference conditions. A comparison of ANN versus the 10th degree polynomial for first example is shown in Fig. 10, while the second example is shown in Fig. 11.

As illustrated the Figs. 10 and 11, the EBaLM-THP was superior to the 10th polynomial as the polynomial essentially serves as a 'moving average' of the nonlinear trend exhibited by the reference condition. The polynomial fitting was unable to mimic either the fluctuating or oscillatory trend in the data. Meanwhile, ANN predicts many to most data points. For the first example, the total error of the 10th polynomial fitting was 2.299016210, while that of ANN 8-4-1 was a smaller 0.329307030. The computed total errors of 10th polynomial for ΔP_c, ΔP_h, and Q were 0.360642000, 1.645394000, and 2.056541000 respectively, whereas those of ANN 8-4-3, which had the poorest performance, were only 0.088750320, 0.182462560, and 0.093324986 for the three different output variables modeled (ΔP_c, ΔP_h, and Q, respectively). The ANN approach clearly yields more representative results. Thus, if the parameter space for an engineered thermal system component is designed to work under nonlinear load and/or in conjunction with significant changes in the equation of state, the above described ANN approach can facilitate the design and analysis tasks.

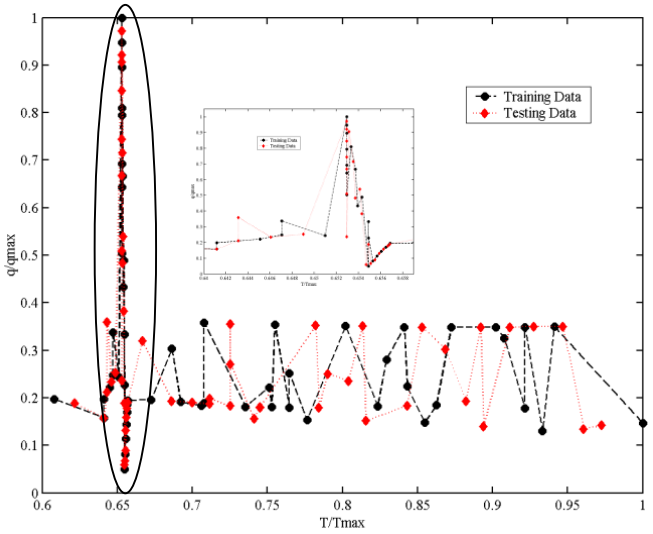


Fig. 6. Training and testing datasets.

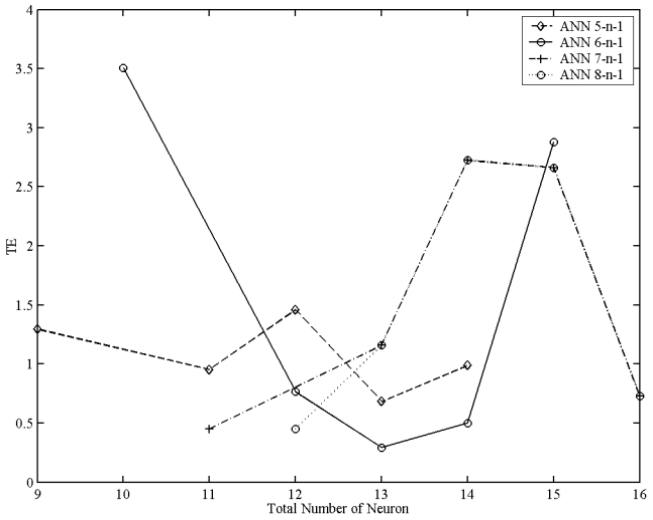


Fig. 7. Comparison of total error versus total number of neuron for each group of ANN architectures.

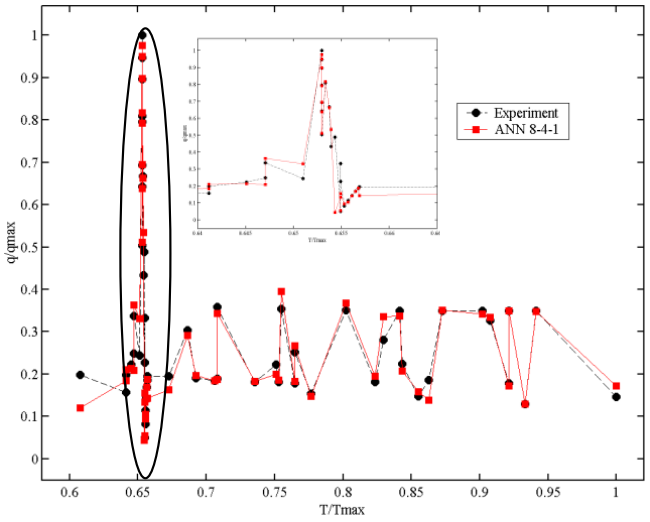


Fig. 8. Comparison of experimental data versus ANN outputs for ANN 8-4-1.

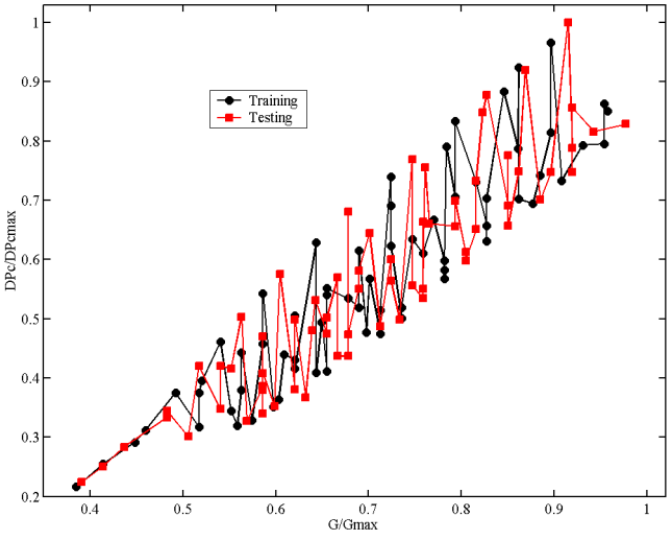


Fig. 9. Plots of training data and testing data for cold-sided pressure drop.

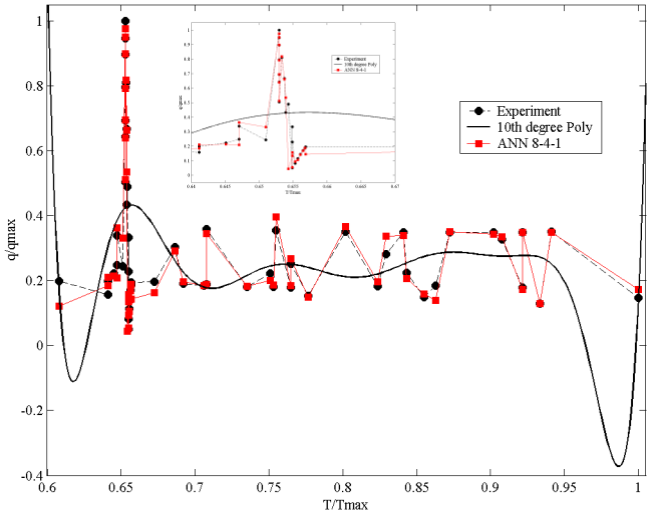


Fig. 10. The comparisons of ANN 8-4-1 versus 10th order PF.

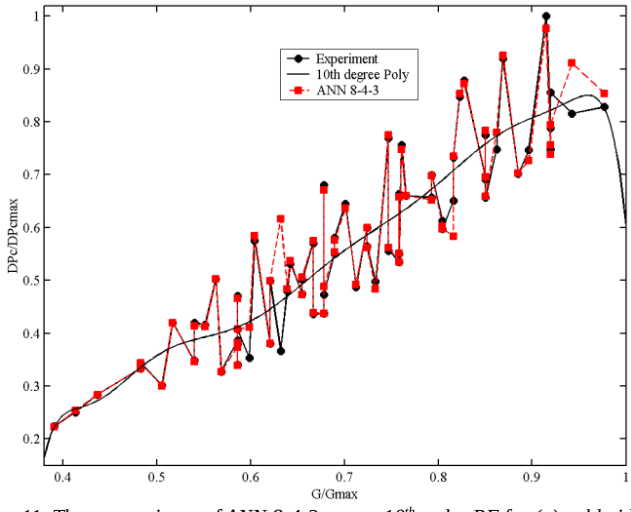


Fig. 11. The comparisons of ANN 8-4-3 versus 10th order PF for (a) cold-sided pressure drop.

It can be concluded that even though the reference data was fluctuating and oscillatory, the neural network was able to follow these characteristic changes. It is evident that the advantage of neural network approach is the network's ability to learn the dataset. In contrast, a polynomial fit was at best able to follow the stepwise average of the oscillatory nature of the reference dataset. Thus, at each step, it failed to fully capture nature of both the nonlinearity and noise contained in experimental data. In fact, the total error of a 10th order polynomial fit was one to three orders of magnitude larger than that associated with the neural networks. As illustrated by Table 3, ANN 8-4-3 architecture showed two orders of magnitude better *TE* than the 10th order polynomial (0.093324986 vs. 2.056541) for the output Q.

Table 3. Total Error comparisons, ANN vs. polynomial fitting

Network vs. Polynomial	Total Error
ANN 8-4-1 (q''_w output)	0.32930703
Polynomial Fitting of 10th order (q''_w output)	2.29901621
ANN 8-4-3 (DPc output)	0.08875032
Polynomial Fitting of 10th order (DPc output)	0.360642
ANN 8-4-3 (DPh output)	0.18246256
Polynomial Fitting of 10th order (DPh output)	1.645394
ANN 8-4-3 (Q output)	0.093324986
Polynomial Fitting of 10th order (Q output)	2.056541

Case study 2

The second study improves EBaLM-THP via EBaLM-OTR technique for ANN architecture selection for PCHE modeling [Wijayasekara 2011].

The main motivation for the work in this study was the frequently overlooked issue of overfitting. Researchers typically focus on minimizing the training (learning) error, which may adversely impact generalization capability of the model. In addition to main criterion for optimal architecture selection, the EBaLM-OTR looks into the testing error, while the third criterion ensures the confidence into the chosen architecture, i.e., the standard deviation of the training error.

In order to perform exhaustive data generalization, the EBaLM-OTR technique for ANN architecture selection for PCHE modeling used flavor of *k*-fold (*5*-fold in this case) cross validation which divided the dataset into *5* similarly sized folds (one fold for testing, one for validating, and three folds for training). Thus the dataset Y is divided into three portions containing P_{tr} number of training data patterns (Y_{tr}), P_{te} number of testing data patterns (Y_{te}), and P_{va} number of validating data patterns (Y_{va}), where:

$$P = P_{tr} + P_{te} + P_{va} \tag{10}$$

$$Y = \{Y_{tr} \cup Y_{te} \cup Y_{va}\} \tag{11}$$

The dataset Y contains P number of data patterns y_i and y_i is an $n+m$ dimensional vector containing n number of inputs and m number of desired outputs, i.e.,

$$y_i = \{x_1, x_2, \ldots, x_n, d_1, d_2, \ldots, d_m\} \quad (12)$$

where x_i are inputs and d_i are desired outputs.

After the training was completed, the MSE of the training set, MSE_{tr} (similarly for the testing and validation set) was calculated using

$$MSE_{tr} = \sum_{p=1}^{P_{tr}} (d_p - o_p)^2 . \quad (13)$$

Fig. 12 shows the pseudo code for evaluation of one ANN architecture. At step1 the network is initialized using a random set of weights. Then at step 2 and 3 the dataset is reordered randomly and divided into training, testing and validating sets. At step 6 the network is trained using the training dataset (Y_{tr}). The training is performed until the network achieves a predefined error or reaches a certain number of iterations (step 5). The number of iterations was set to 500 and the goal error was set to 10^{-5} [Wijayasekara 11].

```
1.  initialize the network with a set of random weights
2.  reorder the data points randomly
3.  divide the dataset into k folds
4.  FOR all k
5.  WHILE training error < 10^-5 OR 500 iterations
6.  train the network using Ytr
7.  END WHILE
8.  test the network using Yte
9.  validate the network using Yva
10. calculate training MSE
11. calculate testing MSE
12. calculate validating MSE
13. END FOR
```

Fig 12. Pseudo code of evaluation of one network architecture

For each architecture Steps 1 through 13 were repeated Q times, and mean and standard deviation MSE for of training, testing and validation were calculated. This process was repeated for all the architectures tested. Fig. 13 shows the complete algorithm in a flow chart format.

The goal of the analysis was to select the architecture that can fulfill both Criteria 1, architecture that can model the data with highest accuracy, and Criteria 2, architecture that achieves the best data generalization.

For the first goal, *Conf*, (the architecture that conforms to the data best), the authors used the architecture with the lowest sum of errors for both training and validating, i.e., the architecture that predicts the experimental data with lowest errors. The mean MSE of the testing set is used to further evaluate the conforming capability of the network.

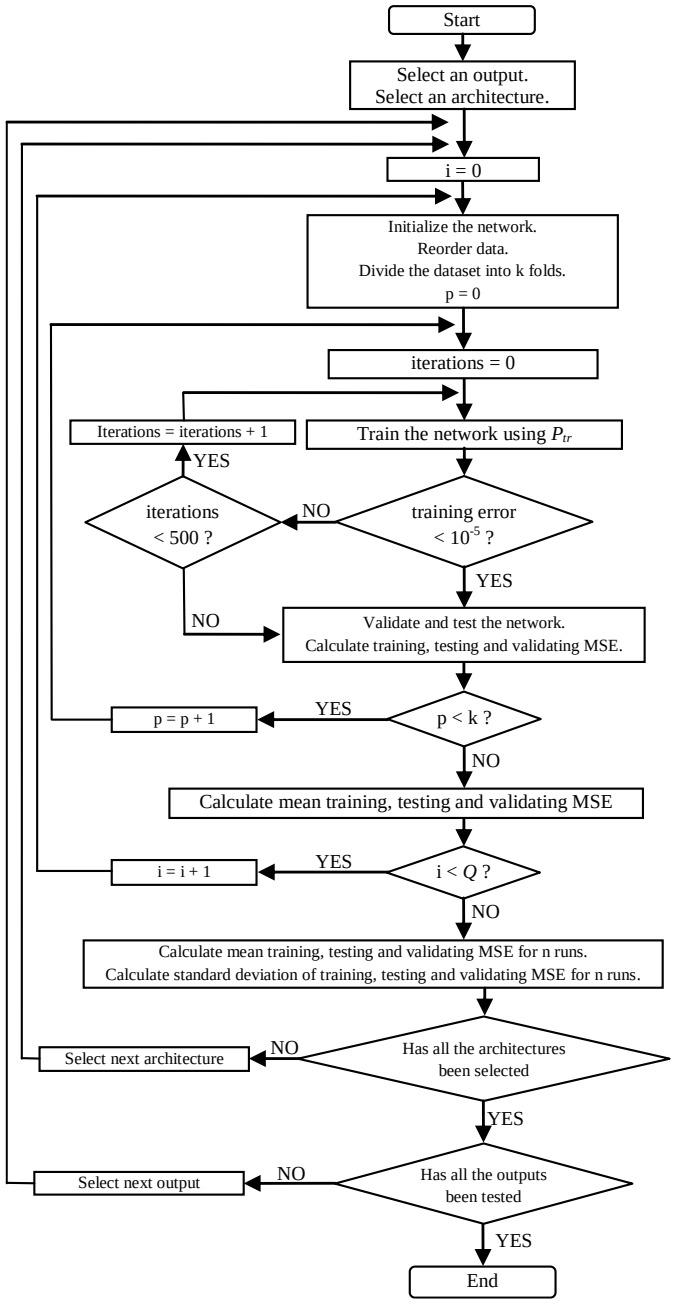


Fig 13. Flowchart of evaluation of network architecture

For the second goal, *Gen*, (the most generalized architecture), the architecture with smallest difference between training and validating mean MSE was chosen. Therefore, to evaluate the generalization capabilities of the architecture, the difference between mean MSE values for training and validating were compared.

While the mean MSE was used to evaluate architectures based on conforming capability and generalization capability, the standard deviation of MSE was used to evaluate the architecture's stability in producing results (the architectures with high standard deviations were eliminated).

In this case study, PCHE analysis was based on the dataset by Ishizuka [2005]. Because of different scales and nature of the measurements and values of each output dimension, neural networks were designed so that each network produced one output. Thus, each network had five input and one output neuron, and three different architectures were selected, one for each output.

Architectures were labeled as *p:q:r:s* or *p:q:s* depending on the number of hidden layers, where *q* and *r* were numbers of neurons in hidden layers, and *p and s* were numbers of neurons in input (output) layers. For example, architecture 5:3:4:1 had two hidden layers containing three and four neurons in each, with the input layer containing five neurons and the output layer with one neuron. Similarly 5:4:1 is an architecture with an input layer containing five neurons, one hidden layer with four neurons and an output layer with one neuron.

The architectures that were considered ranged from 5:1:1 to 5:9:9:1 (pretesting with fewer neurons produced too large errors, while the larger architectures were too computationally intensive for practical use). Each architecture was tested 10 times with five-fold cross-validation.

The improvements of EBaLM-OTR for architecture selection compared to EBaLM-THP can be elaborated as follows.

First, since the EBaLM-OTR algorithm introduced the validation data set (training validation), it was immune to over-training (minimized over-training and maximized generalizability). Therefore, compared to the EBaLM-THP algorithm, architectures derived from EBaLM-OTR are more robust and less prone to errors in real-world applications and when dealing with noisy data.

Secondly, since EBaLM-OTR trained and tested each architecture multiple times with the standard deviation of MSE as a selection criterion, the architecture selected by the EBaLM-OTR algorithm was more consistent in producing low error outputs, compared to the EBaLM-THP algorithm.

Thirdly, the dependency of the LM algorithm on the initial weight set was alleviated by running the algorithm multiple times using different randomly initialized weights each time.

Such improvements allowed the EBaLM-OTR algorithm to be used to derive optimal ANN architecture for any problem that requires a noise and initialization resilient algorithm, while at the same time featuring consistent prediction.

The Graphs in Figs. 14 and 15 document the top 15 architectures with the highest conforming capability (*Conf*) and generalizing capability (*Gen*), respectively (cold sided pressure drop). The architectures were sorted in the ascending order of $mMSE_{va}$ + $mMSE_{tr}$ or "Validating + Training Error" for best conforming, or in the ascending order of $mMSE_{va}$ to $mMSE_{tr}$ (Validating—Training Error) for best generalization capability. In order to select the optimal architecture for cold sided pressure drop, the architecture with the best combined score should be selected.

Table 4 lists the top architectures for cold sided pressure drop. The "rank" is the sum of the positions of architectures in each figure (for example, architecture 5:3:1:1 appears in Fig. 14 in the 2nd position and in Fig. 15 in the 7th position, thus has a rank of *2 + 7* = 9). The positions of the architectures in each graph indicate a measure of performance. The table also list the training mean standard deviations indicating the stability of the architecture.

While 5:6:1:1 architecture scores the best for the *Conf* feature (Fig. 14), it is only at the 10th place based on the *Gen* (Fig. 15). On the other hand, not so great conforming–wise architecture (5:3:1:1), proves to be of much better generalization capability, scores overall the best (rank = 9). Hence, the best architecture from the point of both criteria, it is the optimal architecture, while the best architecture from the aspect of generalization, 5:1:5:1, performs so poorly relative to conforming criteria, it is off the chart in Fig. 14.

As mentioned before, EBaLM-OTR looks into testing error as the second selection criterion for optimal architecture selection. Hence, out of the three top scored architectures, 5:2:1:1 was selected as the optimal architecture to model cold sided pressure drop. Finally, the third criterion, standard deviation of training error, was considered for all three architectures. This criterion did not produce a significant difference, so the 5:2:1:1 architecture remained the best choice.

The results of the similar procedure, applied to hot sided pressure drop and heat transfer, are shown in Table 5, while comparisons against the EBaLM-THP algorithm are illustrated in Table 6.

The main instrumentation and control challenge entails several sub challenges such as: identification of hidden but inherent relationships among nonlinear, highly dimensional process variables; knowledge discovery; and relevant data identification in massive data sets created by fusion of large number of sensors. These challenges are anticipated to be successfully tackled using CITs, tailored and enhanced to solve diverse, heterogeneous massive data.

IV CONCLUSION

Artificial neural networks in small modular reactor modeling present a viable mechanism for modeling and optimization. Two case studies based on the previous application of hybrid ANN algorithm—EBP and LM—applied to PCHE modeling demonstrated the effectiveness of ANNs (EBaLM-THP algorithm). Introducing EBaLM-OTR improved generalization and alleviated over training while introducing testing error and standard deviation of error, which can improve modeling error up to two orders of magnitude. Computational Intelligence will serve as a research and modeling platform to improve the sustainability and performance of next generation small modular reactors while maintaining data security for sensitive information.

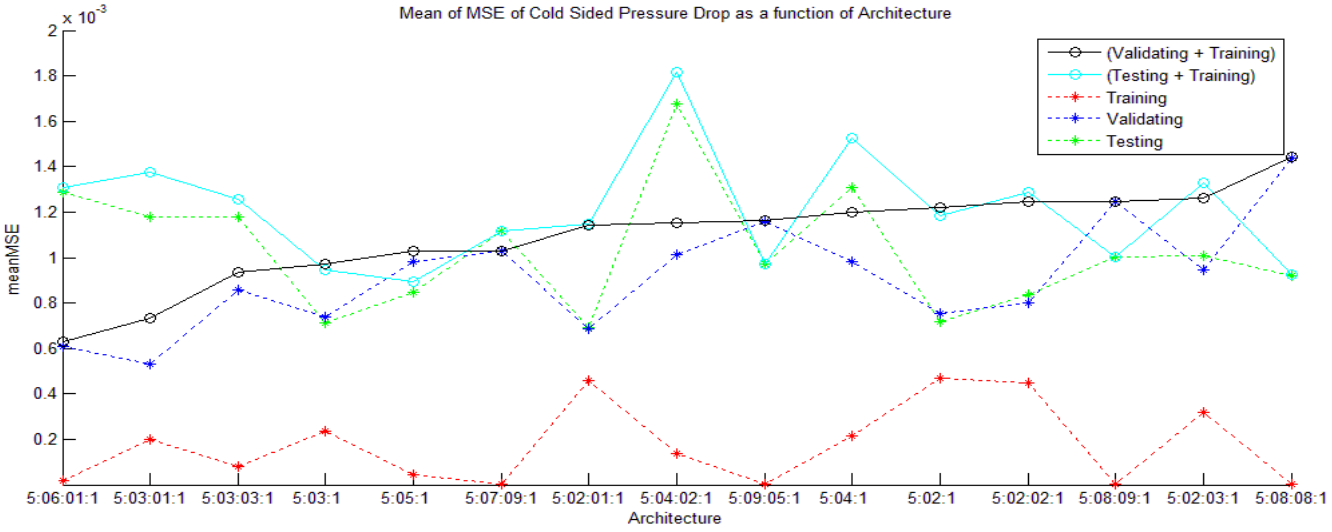


Fig. 14. Conforming capability of network architectures for cold sided pressure drop

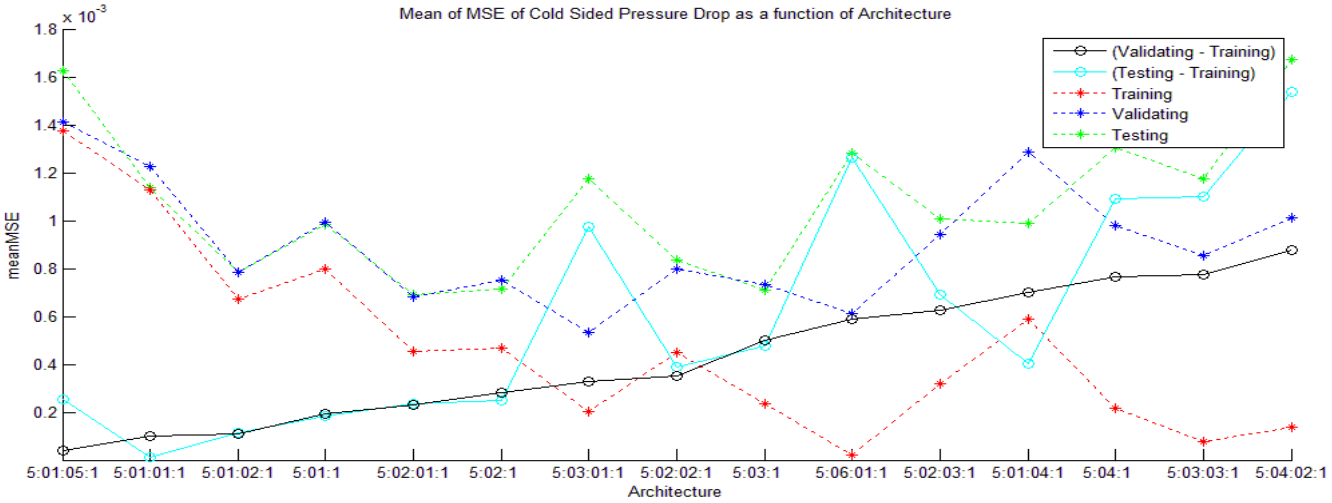


Fig. 15. Generalization capability of network architectures for cold sided pressure drop

Table 4. Selection Criteria for The Optimal Architecture for The Cold Sided Pressure Drop

Architecture	*Conf* (Validation +Training)	*Gen* (Validation -Training)	Rank	Training mean MSE
5:3:1:1	0.733 x 10^{-3}	3.31 x 10^{-4}	9	0.0012 ± 0.0072
5:6:1:1	0.6317 x 10^{-3}	5.903 x 10^{-4}	11	0.0013 ± 0.0067
5:2:1:1	1.142 x 10^{-3}	2.287 x 10^{-4}	12	0.00069 ± 0.0039
5:3:1	0.9715 x 10^{-3}	5.014 x 10^{-4}	13	0.00071 ± 0.0039
5:3:3:1	0.9338 x 10^{-3}	7.78 x 10^{-4}	17	0.0012 ± 0.0042
5:2:2:1	1.246 x 10^{-3}	3.501 x 10^{-4}	20	0.00084 ± 0.0071
5:4:1	1.201 x 10^{-3}	7.662 x 10^{-4}	23	0.0013 ± 0.0227
5:4:2:1	1.153 x 10^{-3}	8.775 x 10^{-4}	23	0.0017 ± 0.0074
5:2:3:1	1.261 x 10^{-3}	6.257 x 10^{-4}	25	0.001 ± 0.0089
5:2:1	1.1219 x 10^{-3}	2.838 x 10^{-4}	27	0.00072 ± 0.0044

Table 5. Performance of The Selected Architectures

Architecture	Cold Sided Pressure Drop			Hot Sided Pressure Drop			Heat Transfer		
	$mMSE_{tr}$	$mMSE_{va}$	$mMSE_{te}$	$mMSE_{tr}$	$mMSE_{va}$	$mMSE_{te}$	$mMSE_{tr}$	$mMSE_{va}$	$mMSE_{te}$
5:2:1:1	0.4564×10^{-3}	0.6851×10^{-3}	0.6920×10^{-3}	0.1324×10^{-3}	0.2022×10^{-3}	0.2054×10^{-3}	0.4473×10^{-4}	0.7160×10^{-4}	0.6992×10^{-4}
5:1:5:1	0.14×10^{-2}	0.14×10^{-2}	0.16×10^{-2}	0.1251×10^{-3}	0.1935×10^{-3}	0.1917×10^{-3}	0.3321×10^{-4}	0.7575×10^{-4}	0.7441×10^{-4}
5:7:9:1	0.9667×10^{-6}	0.1×10^{-2}	0.11×10^{-2}	0.9667×10^{-6}	0.8361×10^{-3}	0.1×10^{-2}	0.85×10^{-6}	0.2395×10^{-4}	0.3245×10^{-4}

Table 6. Mean Absolute Error of EBaLM-OTR Compared to EBaLM-THP

Variable	EBaLM-THP Architecture1	EBaLM-THP Architecture2	EBaLM-OTR
Cold Sided Pressure Drop (kPa)	4.5372	6.1291	1.2515
Hot Sided Pressure Drop (kPa)	1.1999	1.0795	3.2232×10^{-1}
Heat Exchange (kW)	1.0703×10^{-1}	7.8027×10^{-2}	9.9899×10^{-3}

ACKNOWLEDGMENTS

We would like to thank Vivek Utgikar from University of Idaho and Mike Patterson of Idaho National Laboratory for providing positive reinforcement for our efforts.

V REFERENCES

Boroushaki, M., Ghofrani, M.B., Lucas, C., 2005, "Simulation of nuclear reactor core kinetics using multilayer 3-D cellular neural network". *IEEE Trans. Nucl. Sci.*, 52, 719–728.

Diaz, G. and Sen, M., 1999 Simulation of Heat Exchanger Performance by Artificial Neural Network", *HVAC R Research*, 5, 3, 195–208.

Ermis, K., 2007"ANN modeling of compact Heat Exchanger" *Int. J. Energy Res*

Eryurek, E., Upadhyaya, B.R., 1990. "Sensor validation for power plants using adaptive backpropagation neural network." *IEEE Trans. Nucl. Sci.*, 37, 1040–1047.

Garg, A., Sastry, P.S., Pandey, M., Dixit, U.S., Gupta, S.K., 2007. "Numerical simulation and artificial neural network modeling of natural circulation boiling water reactor." *Nucl. Eng. Des.* 237, 230–239.

Guanghui, S., Morita, K., Fukuda, K., Pidduck, M., Dounan, J., Miettinen, J., 2003. "Analysis of the critical heat flux in round vertical tubes under low pressure and flow oscillation conditions. Applications of artificial neural network." *Nucl. Eng. Des.*, 220, 17–35.

Guo, Z., Uhrig, R.E., 1992. "Nuclear power plant performance study by using neural network." *IEEE Trans. Nucl. Sci.*, 39, 915–918.

He, S., Jiang, P.X., Xu, Y.J., Shi, R.F., Kim, W.S., and Jackson, J.D., 2005, "Computational Study of Conviction Heat Transfer to CO_2 at Supercritical Pressures in A Vertical Mini Tube." *Int. J. Therm. Sci.*, 44, 521–530.

Ishizuka, T.,Kato, Y., Muto, Y., Nikitin, K., and Tri., N.L., 2005. "Thermal-Hydraulic Characteristics of A Printed Circuit Heat Exchanger in Supercritical CO_2 Loop." *The 11th Int. Topical Meeting on Nucl. Reactor Thermal-Hydraulics, Pops' Palace Conference Center, Avignon, France, October 2–6*.

Jia, R., Sunden, B., 2003. "Optimal design of compact heat exchangers by an artificial neural network method," HT2003, *ASME summer heat transfer conference*, July 21–23, Las Vegas, Nevada.

Hun Kim, Hee Cheon No, Jeong Ik Leea, Byong Guk Jeona, 2009. "Thermal hydraulic performance analysis of the printed circuit heat exchanger using a helium test facility and CFD simulations," *Nuclear Engineering and Design*, 239, 11, November 2009, 2399–2408.

Levenberg, K., 1994"A Method for the Solution of Certain Problems in Least Square." *Quart. Appl. Math.*, 2, 164–168.

Manic, M., 2011 "Data Mining," *The Industrial Electronics Handbook*, 2nd ed.,Vol. 3, Intelligent Systems, CRC PressTaylor & Francis Group, Editors Irwin J. D., Wilamowski, B., Print ISBN: 978-1-4398-0283-0, eBook ISBN: 978-1-4398-0284-7, pp.30-1:30-15, Feb 28, 2011.

Manic, M., (i) 2011. "Neural Networks Design course, lecture notes," *University of Idaho, Spring 2011, URL from June 2011: http://husky.if.uidaho.edu/ee578s11/.*

Marquardt, D., 1994, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." *SIAM J. Appl. Math.*, 11, 431–441.

Mazrou, H., 2009. "Performance improvement of artificial neural networks designed for safety key parameters prediction in nuclear research reactors," *Nuclear Engineering and Design*, 239, 10, October 2009, 1901–1910.

Pacheco-Vega, A., 2001. "Neural Network Analysis of Fin-Tube Refrigerating Heat Exchanger with Limited Experimental Data." *Int. J. Heat Mass Transf.*, 44, 763–770.

Pacheco-Vega, A.J., 2002, "Simulation of Compact Heat Exchangers Using Global Regression and Soft Computer"

Ph.D. dissertation, Depart. Aero. and Mech. Eng., Univ. of Notre Dame, Notre Dame, Indiana.

Fast, M., and Palme, T., 2009, "Application of artificial neural networks to the condition monitoring and diagnosis of a combined heat and power plant," *Energy*, 35, June 2009, 1114–1120.

Peng, H., and Ling, X., 2007, "Optimal Design Approach for the Pate-Fin Heat Exchanger using Neural Networks Cooperated with Genetic Algorithms" *Applied Thermal Engineering*, 28, 5-6, 642–650.

Ridluan, A., Manic, M., Tokuhiro, A., 2009."EBALM-THP – A Neural Network thermal-hydraulic Prediction Model Of Advanced Nuclear System Components," *Elsevier, Int. Journal on Nuclear Engineering and Design*, 239, 2, February 2009, 308–319.

Roh, M.-S., Cheon, S.-W. and Chang, S.-H., 1991. "Thermal Power Prediction of Nuclear Power Plant Using Neural Network and Parity Space Model." *IEEE Trans. Nucl. Sci.*, 38, 866–872.

Rumelhart, D.E., and McClelland, J.L., 1986. "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," *Vol.1, Cambridge, MA, MIT Press.*

Schultz, R.R., Nigg, D.W., 2004. "Next Generation Nuclear Plant-Design Methods Development and Validation Research and Development Program Plan", *INEEL/EXT-04-02293, Idaho National Engineering and Environmental Laboratory*, September 2004.

Vaziri, N., Hojabri, A., Erfani, A., Monsefi, M., Nilforooshan, B., 2007. "Critical heat flux prediction by using radial basis function and multilayer perceptron neural network: a comparison study." *Nucl. Eng. Des.*, 237, 377–385.

Waheda, M.E., and Ibrahim, W.Z., 2010. "Neural network and genetic algorithms for optimizing the plate element of Egyptian research reactor problems", *Nuclear Engineering and Design*, 240, 1, January 2010, 191–197.

Werbos, P.J., 1974. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences." *PhD thesis*, Harvard University.

Werbos,P.J., 1994. "The Roots of Backpropagation". New York: *Johns Wiley & Sons.*

Wijayasekara, D.S., , Manic, M., Sabharwall, P., Utgikar, V., 2011 "Optimal artificial neural network architecture selection for performance prediction of compact heat exchanger with the EBaLM-OTR technique." *Nucl. Eng. Des.*, doi:10.1016/j.nucengdes.2011.04.045.