RESEARCH ARTICLE OPEN ACCESS

# Balancing the Complexity of Architecture and Generalization of Soft-Computing Model in Predicting the Properties of Composite Preforms

Dr. P. Radha
Associate Professor
MCA Department
Mepco Schlenk Engineering College, Sivakasi
Tamil Nadu - India

**ABSTRACT**
The novel strategy proposed in this paper is used to reduce the complexity of architecture of soft-computing model like neural network with high accuracy in predicting the outputs. It further improves the recognition power of neural network while handling raw-data with highly non-linear, more interrelated, noisy and MAR (Missing At Random) values. The bias term was slightly modified in MRBNN (Modified Radial Basis Neural Network) to improve the generalization of over-fitting problems. The architecture of network model was balanced with the network generality in Powder metallurgy Lab for predicting the deformation and strain hardening properties of AI-Fe composite preforms.

***Keywords:-*** soft computing, Radial Basis neural network, composite preforms

## I. INTRODUCTION

Soft computing, as opposed to conventional "hard" computing, is a technique that is tolerant of imprecision, uncertainty, partial truth and approximation. Its methods are based on the working of the human brain and it is commonly referred to as Artificial Intelligence (AI). The action of AI is similar to the human brain which is capable of arriving at valid conclusions based on incomplete and partial data obtained from prior experience. The soft computing methods are robust and low cost. The application of soft-computing tools in the material engineering was analyzed in the early research [1]. In this article, the soft-computing tool like neural networks was applied in Powder Metallurgy area to process the properties of metal powders. The soft-computing based Simulation of powder metallurgical preforms may avoid lab experiments involving dangerous materials and hence prevent risky consequences. This model is not only avoids expensive experiments but also evade handling dangerous materials that cause severe damage to environment.

As per the existing models adopted in the previous research [2-5] with neural networks, the following factors were identified.

- Little number of outputs is possible from more number of inputs: More inputs are used to derive one or two outputs. In this case, it is not necessary to use the soft-computing approach.The number of inputs is more than thrice amout of outputs.
- No strategy, in fixing the correct combination of input features– arbitrarily & manually:
- Many input features are involved in designing the soft-computing model. The relevant input feactures can be selected only by the experience of technicians in powder metallurgy Lab. Also, for each set of combination, this model gives different results. To overcome this problem, the standard approach is used to select the input features in this paper.
- Applied more number of training samples: Generally, RBF needs more samples for proper training. In terms of thousands, training samples were preferred for training in previous research.
- Spent more number of hidden neurons: Due to more training samples, the size of hidden layer is high in case of exact interpolation of earlier RBF models.

- Individual neural network was used for each characteristic of material: The eariler model can not predict the various deformation and strain hardening properties simultaneously. For each output, a separate network model was used.
- Handled with only linear outputs: The developed models can not predict the complex parameters.
- No more efficient neural network design in powder metallurgy field in handling composite preforms: The effeciency was considered in desinging earlier models of powder metallurgy field.
- No more networks in predicting the Al-Fe composite preforms with null entries: The previous models did not consider about the null entries.
- No concrete method in dealing over-fitting problems. [6-9]: Due to the poor capability of neural network, the validation error with dependent samples, may be less. But the tesing error with independent samples will be high.

Based on the above factors in the earlier research, the scope of this work is:

- To investigate novel neural network strategy to predict more network outputs data in terms of both linear and nonlinear from limited relevant inputs: The proposed neural network model can be helpful to process both simple and complex parameters.
- To optimize the network architecture (in terms of hidden layer size and no. of training samples) & training time: Less amount of training time may be spent in proposed neural network with minimum hidden layer size.
- To select the relevant input combinations using standard procedure: The new model can select the possible relevant input features with standard strategy.
- To process highly non-linear, interrelated, noisy / null data: The raw data has more complex parameters. The output of some parameters will be used as input to another parameters. Also, some parameters have more null entries.
- To improve the accuracy of network: The predictabilty of neural network should be improved along with the above objectives.

## 2. Raw data analysis

While analyzing the raw data, it had the following properties:

- Interrelated Data*:* Difficult to specify input combinations for each output category due to interrelation among the parameters. The certain output parameter should be fed as input to derive the other output parameters. The strain factor and Poisson's ratio based on contact & bulged parameter are computed using axial strain and hoop strain [9]. The Poisson's ratio is computed using conventional hoop strain and hoop strain. Hydrostatic stress is computed using axial and hoop stress.
- Noisy data: While measuring the parameters of raw-data using physical instruments in P/M Lab, it is possible to have the addition of noisy data with the raw-entries.
- MAR (Missing At Random): More rows of some parameters were NULL entries. It is treated as MAR since the existing value does not depend on only missing data. Hence it is away from the MNAR (Missing Not At Random).In the raw data base Table 1, the empty cell of $n_i$ and $k_i$ parameters can be derived from the difference between two subsequent rows of known column value. In each data set, for the every first row, there is no previous row value. Hence these entries were made as zero.
- Highly non-linear data: More parameters have non-linear relationship with one another. Figure 1(a) shows the linear relationship between axial strain and hoop strain. Figure 1 (b) shows the non-linear relationship between variation of stress (axial, hoop and hydrostatic stress) and axial strain.

**Fig. 1(a)** Linear relationship

**Fig. 1(b)** Non-linear relationship

**Table 1** Raw data by physical experiments

### 3.0 Conventional and Proposed Model

The proposed model can give more number of both linear and non-linear output parameters from limited data patterns with limited extracted features. It improves the recognition power of NNs in working data with highly non-linear, more interrelated, noisy and NULL entries. The strength of hidden layer is reduced using clustering method and the training with bias term is slightly modified to improve the generalization network. It adopts linear function as an efficient radial basis filter. Generally, the needed inputs for network are being selected using the working experience of the developers. Hence it is impossible to select the correct input combination for each output parameter. This system decides the relevant input features .The proper strategy should be decided in normalizing the NULL entries of raw data by zeros or mean of the output parameter or the NULL entries may be completely removed from the raw database.

The proposed network model was designed to predict the deformation characteristics of AI-Fe composite preforms used in powder metallurgy (P/M) Lab such as axial strain ($\varepsilon_z$), hoop strain ($\varepsilon_\theta$) conventional hoop strain ($\varepsilon_\theta$ ·), strain factor ($S$), Poisson's ratio based on contact diameter ($\nu$), Poisson's ratio based on contact & bulged diameter ($\psi$), axial stress ($\sigma_z$), hoop stress ($\sigma_\theta$) and hydrostatic stress ($\sigma_M$) of aluminium- iron composite preforms used in powder metallurgy lab. In addition to that, the value of strain hardening coefficients such as instantaneous strength coefficient ($k_i$) and instantaneous strain hardening exponent ($n_i$) are also simulated to find the effect of the percent of iron content on formability using the input parameters such as load, aspect-ratio, fractional density, iron content and lubricant. The correlation between experimental and predicted value from simulated model was compared for error calculation. This system was compared with the toughest non-linear benchmark problem, like XOR problem and the related problems of powder metallurgy field.

### 3.1 Radial basis Neural Network with NADP

Radial Basis Neural Networks (RBNN) offers a powerful framework for representing non-linear mappings from several inputs to one or more outputs. RBFs are feed-forward networks consisting of a hidden layer of radial kernels and an output layer of linear neurons [1, 9]. The connection between input and hidden layer does not use the weighted sum of inputs. The output of the hidden layer represents basis functions, which are determined by the distance between the network input and the center of the basis function. As the input moves away from a given center, the neuron output drops off rapidly to zero. The output layer of RBF network is linear and produces a weighted sum of the outputs of the hidden layer. The neurons in the RBF network have localized receptive fields because they only respond to inputs that are close to their centers.

The exact interpolation called Neuron at data point (NADP) is used to map every point in the input pattern to the output layer. Formally, the exact interpolation of Q data points in a multi-dimension space require all the D dimensional input vectors $x^k = \{ x_i^k, i= 1, 2 \ldots D\}$ to be mapped onto the corresponding target output $y^k$ [11] where D is the size of Input layer, The goal is to find the $f$ function such that

$$f(x^k) = y^k \qquad \forall k \quad = 1 \ldots\ldots Q \tag{1}$$

where Q represents number of training samples. This approach requires $Q$ amount radial basis functions. The generalization performance of RBF network relates to its prediction capability on independent test data [1].

The algorithm given below was used to interpolate the source data exactly:

***Step 1:*** Choose the free parameters like the spread factor, number and values of centers and the type of radial basis function. In this model, all training samples are considered as centers of RBF.

***Step 2:*** The training pattern with D amount of features is applied to input layer X, whose size is equal to D .Then each input node $(X_i ; i=1,2,..D)$ sends the input data to the hidden layer.

***Step 3:*** At hidden layer, the distance between the input layer $X$ and the centers is calculated

by the formula:

$$\| X- \mu_j \| = \sqrt{\sum_{i=1}^{i=D} \sum_{j=1}^{j=Q} (x_{i,j} - \mu_{j,i})^2}$$

(2)

where, the entire training samples is considered as center set μ in NADP model and $\mu_j$ is j[th] center point.

***Step 4:*** The activation of hidden unit (ϕ)is determined by distance between the input vector and centers using any one of the RBF filter $f$

$$\phi_j (X) = f(\| X - \mu_j \|)$$

(3)

The filter $f$ may be in the form of Gaussian function, cubic function, linear function, Multi Quadric function or inverse Multi-quadric function.

***Step 5:*** The activation of the output unit is determined by dot product between the hidden activation vector and weight vector [12]. For convenience an additional basis function $\phi o$ with constant activation value of '1' can be used with unknown weight $W_0$.

$$y_k = \sum_{j=1}^{Q} W_{jk} \phi_j (X) + Wo\phi o$$

(4)

### 3.2 Modified Radial basis Neural Network

Every input point must appear as part of the system used to model the data without averaging or smoothing in exact interpolation. To solve the system of eqn. (4) easily, matrix form can be followed for deriving the unknown values from the known values.

$$Y = (y_1, .... y_Q)^T$$

(5)

$$W = (w_1, ..... w_Q)^T$$

(6)

$\phi$ is a $Q$ x $Q$ matrix computed entirely from the data points $X$

$$\phi = \begin{pmatrix} \phi(\| X_1 - \mu_1 \|) ..... \phi(\| X_1 - \mu_Q \|) \\ \vdots \\ \phi(\| X_Q - \mu_1 \|) ..... \phi(\| X_Q - \mu_Q \|) \end{pmatrix}$$

(7)

Since $\phi$ is symmetric, that is

$$Y = \phi^T W = \phi W$$

The bias term is added with the output of hidden layer in improving the accuracy of output $Y$.

The activation function value $Q_0$ of bias term is assumed as one in Equation 4. The unknown weight factor $W_0$ is added with the summation factor and the matrix form is formed as:

$$\phi W b = Y$$

(8)

Where $W_b$ is combination of weight W and $W_o$ for the unknown parameter W can be computed by the conventional pseudo inverse of $\phi$ as given below:

$$Wb = \phi^{-1} Y$$

(9)

While testing the independent input samples, $W_b$ is separated as W and $W_o$ to derive the output value Y using the equation (4).

In the proposed MRBFNN algorithm, the unknown weight value for bias term Wo is generated randomly and they are not applied to RBF filter as followed in conventional method. Only the weight W between hidden and output nodes is applied to RBF filter. While training, the weights are computed using the known target value Y, bias weight and $\phi$ functions as given below:

$$W = \phi^{-1} Y - \phi^{-1} W_0$$

(10)

For simulating the network, the calculated W was used to recognize independent testing samples.

### 3.3 Identifying PRNET model

Principal Component Analysis (PCA) involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. It helps to discover or to reduce the dimensionality of the data set and to identify new meaningful underlying variables.

PRNET (PCA-Radial NETwork) model is a combination of PCA and MRBFNN. The raw data set has more number of input parameters in the form of P (Load, *kN*), $H_o$ ( mm) , $H_r$ (mm), $D_o$ (mm), $D_c$ (mm), $D_B$ (mm), Iron content, Lubricant, $\alpha$ , $\pi$ , $\delta$ and fractional density. While using with more input parameters, the network complexity increases in architecture and it will turn in increasing the training time. Hence the selection of limited number of input features is the main issue in designing the architecture of neural network. Using the working experience in P/M Lab, the input layer size may be decided. But the different input combination will provide different result. There is no assurance to select the unique optimal selection among the available input features.

The PCA tool helps in reducing the dimension of input space, which in turn reduces the architecture and training time. Also it provides the unique path for deciding the possible input features. The MRBFNN algorithm with NADP model has main impact in predicting the properties within the defined constraints. The samples were preprocessed and the input features of the network model should be selected in a systematic way with the help of PCA. To improve the learning time of network, the raw data was normalized. The raw data x was mapped as $x_n$ between 0 and 1, as given below:

$$X_n = N_{\min} + (X - \min_x)\frac{(N_{\max} - N_{\min})}{(\max_x - \min_x)} \quad (11)$$

where the $\min_x$ and $\max_x$ are the minimum and maximum value of x respectively. The $N_{min}$ and $N_{max}$ are the minimum and maximum value of $x_n$ respectively. Since the $N_{min}$ is zero and $N_{min}$ is one , the above equation was reduced to:

$$X_n = \frac{(X - \min_x)}{(\max_x - \min_x)} \quad (12)$$

Again the normalized network was denormalized by:

$$x = (x_n + \min_x)(\max_x - \min_x) \quad (13)$$

Since the problem seems to be of MAR type, different cases were considered in filling the empty slot in the parameters. The mean of the parameter, which has empty cells, was derived and these blank cells were filled by this mean. Also the network was trained by substituting the empty cells by zeros. The MRBFNN model was used to train the network and finally the output is denormalized for the result analysis. The error is measured in the form of correlation Coefficient (R) and Average Absolute Relative Error (AARE) percentage.

### 4.0 Results and discussions

The proposed RBF model was implemented by the package MATLAB 2010 and BPN model was implemented by C++ compiler- gcc 4.1.2 in Linux environment. The network error was measured in terms of correlation coefficient and (AARE) Average Absolute Relative Error %.

### 4.1 Effect of BPN

The conventional back-propagation neural network was applied. The following thumb rule was used to fix the number of hidden nodes:

3/4 of sum of input and output neurons

12=3/4*(5+11) where 5 is number of input nodes and 11 is number of output nodes. Logistic – sigmoidal filter was used in hidden nodes and pure linear filter was used in input and output nodes. The network was trained with 90 samples and tested with 313 independent samples. It yielded good validation but poor testing results as given in Fig. 2 (a-b).

**Fig.2 (a)** Axial stress –Validation

**Fig.2 (b**) Axial stress –Testing

If the network is not stopped at proper time, it will have the poor generalization. Hence the

generalization tools like early-stopping, regularization and Bayseian network modes were used. These models were not helpful to predict the null parameters (stain hardening properties) as reported in Table 2. Also this table reveals that architecture of BPN system is not balanced with the generalization.

**Table 2 Effect of BPN-Generalizations tools**

It was very difficult to predict the stopping time of network in case of over-fitting problem. From Fig.3, it was noted that the AARE% is more in 3000 than at 2000 epcohes.

**Fig. 3** BPN with Non-Null Parameters

For improving the generalization of BPN, the hidden node size was increased to 30 neurons from 12 and number of training samples was increased from 90 to 225. But it took 14 hours for proper training while Weight regularization and Early stopping took 2 to 3 minutes , Bayesian Network took 10 to 13 minutes , Conventional method BPN (96 samples) took 30 minutes for convergence.

The exiting Back Propagation Network (BPN) model was compared with MRBFNN-(NADP) model as depicted in Fig 4 and observed that the accuracy of MRBFNN-NADP model was higher than BPN. Still the generalization of strain hardening parameters ($n_i$ and $k_i$) *are poor*.

**Fig. 4** Comparison of BPN & MRBFNN-NADP model

### 4.2 Conventional RBFNN Vs Modified RBFNN

Figure 5 compares the output of conventional and modified RBFNN with NADP method. It seems that the MRBFNN model increases the accuracy slightly as compared to conventional model. MRBFNN network model took 0.561471 seconds for training.

**Fig. 5** Comparison of Conventional RBF and MRBFNN

### 4.3 Hybrid network model

The MRBFNN with NADP strategy used 350 hidden nodes. This strength can be reduced by clustering. In the hybrid network model using k-means clustering algorithm, the centroids were selected. This model prefers hidden neurons size, which is smaller than the total training samples *Q*. Using k-means algorithm, the centroids of 200 cluster groups are considered as the hidden nodes of MRBFNN. The MRBFNN was trained with 200 (*k* which is less than *Q=350*) hidden nodes to predict 11 output parameters. Table 4 shows R and AARE% of NADP and Hybrid model with MRBFNN.

**Table 3** Error comparison of NADP and hybrid model in MRBFNN

The clustering algorithm was repeatedly applied with different k values. Except $\psi$, *all* parameters give good results based on the clustering between the range k=150 and 200. Only the hydrostatic stress and instantaneous strain hardening coefficient ($n_i$) can give good results at k=300. Hence it is decided to have optimal k value between the range 150 and 200.

### 4.4 Role of PRNET Model

As mentioned in section 3.3, the PRNET selection with limited input features is best one, than the selection through the working experience. The input features which have the contribution less than 2% are removed from the feature space. From Fig.5, it seems that the PRNET model gives good accuracy for AARE % than the NADP model, where NADP selects the input by working experience in P/M Lab. PRNET predicts both NULL and non-NULL parameters with limited error percentage, due to the input space with limited relevant features. The correlation coefficient R is one in all output parameters during validation. With the help of MRBFNN strategy, the PRNET model proves that the architecture of system is balanced with the network generality.

**Fig. 6** NADP model Vs. PRNET model

## 5. Conclusions

The survey of various kinds of neural network models reported in this article will be helpful

for the researchers to develop the innovative expert systems. MRBFNN – needs less training time, less number of free parameters and increase in the accuracy of the results. PRNET – model gives good results even for NULL parameters. The soft-computing based approach PRNET is useful for balancing the complexity of architecture and generalization of soft-computing model in predicting the properties of composite preforms in case of over-fitting problems. Also it will help the researchers to predict the characteristics of any kind of powder materials even for the nano composites in future.

## REFERENCES


[1] Ali Ghodsi, Dale Schuurmans, Automatic basis selection techniques for RBF networks, International journal on Neural networks, 16 (2003) 809–816.

[2] Cherian R.P., Smith. L.N.,.Midha.P.S, A neural network approach for selection of powder metallurgy materials and process parameters, AI in engineering, 14 (2000) 39-44.

[3] L.A., Honysz. R., Application of artificial neural networks in modelling of normalized structural steels mechanical properties, Journal of Achievements in Materials and Manufacturing Engineering, 32(2009) 37-45.

[4] Bai Guanghui, Meng Songhe, Du Shanyi,Zhang Boming, Liang Jun, Liu Yang, Prediction on the ablative performance of carbon/carbon composites based on artificial neutral network, Acta Materiae Compositae Sinica, (2007) 6-15.

[5] Bottou. L. and Vapnik. V, Local learning algorithms, Neural Computation, 4 (1992) 888-900.

[6] Duin. R. P. W., Super Learning and neural network magic. Pattern Recognition Letters, 15 (1994) 215-217.

[7] Kraaijveld.M. A. and Duin. R. P. W., The effective capacity of multilayer feed forward network classifiers. Proc. ICPR, Israel, B: (1994) 99-103.

[8] Chuanhao Wan, Peter de B. Harrington, Self-Configuring Radial Basis Function Neural Networks for Chemical Pattern Recognition, J. Chem. Inf. Comput. Sci., 39(1999) 1049–1056.

[9] Selvakumar.N., Experimental investigation of Al-Fe sintered composites during cold upsetting, Ph.D., Thesis, 2005.

[10] Satishkumar, Neural Networks, TMH, New Delhi, 2005

[11] Smith. L.N., Modeling PM process using neural networks, Metal Powder Report, 55 (2000) 30–35.

[12] Tamarai Selvi. S., Arumugam. S., Ganesan. L., BIONET: an artificial neural network model for diagnosis of diseases, Pattern Recognition Letters, 21 (2000) 721-740.

[13] Selvakumar. N., Narayanasamy.R., Radha.P., Joseph Davidson. M., Prediction of deformation characteristics of sintered aluminium preforms using neural network. J Model Simulation Mater Sci. Eng, 12 (2004) 611–620.

[14] Selvakumar. N., Ganesan. P., Radha. P., Narayanasamy. R., Pandey. K.S., Modelling the effect of particle size and iron content on forming of Al–Fe composite preforms using neural network, Journal of Materials & Design, 28 (2007) 119–130.

[15] Selvakumar. N., Radha. P., Narayanasamy. R., Ganesan. P., Neural Network model for predicting the strain hardening and densification constants of sintered aluminium preforms, Powder Metallurgy, 47(2004) 261-266.

[16] Poshala. G., Ganesan. P., An analysis of formability of aluminium preforms using neural network, Journal of Materials Processing Technology, 205 (2008) 272-282.

**List of Tables & Figures:**



**Table 1** Raw data by physical experiments

Initial height ($H_0$) = 9.43 *mm* Initial density ($\rho_f$) = 2.60 *g/cc* Initial diameter ($D_0$) = 20.30 *mm*
Weight in air ($w_a$) = 7.88 *g* Lubricant = $MoS_2$ Aspect ratio = 0.50

| Load *kN* | $H_f$ *mm* | $D_{CT}$ *mm* | $D_{CB}$ *mm* | $D_C$ *mm* | $D_B$ *mm* | $W_w$, *g* | $\rho_f$, *g/cc* | $\left(\frac{\rho_f}{\rho_{th}}\right)$ | $\varepsilon_z$ | $\varepsilon_\theta$ | $\varepsilon_{\theta'}$ | $e^{(\varepsilon_z-\varepsilon_\theta)}$ | $\psi$ | $\nu$ | $\sigma_z$ *MPa* | $\sigma_\theta$ *MPa* | $\sigma_m$ *MPa* | $n_i$ | $k_i$ *MPa* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.0 | 9.42 | 20.31 | 20.31 | 20.31 | 20.31 | 3.03 | 2.60 | 0.93 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 15.41 | 7.68 | 2.57 | -- | -- |
| 10.0 | 9.39 | 20.32 | 20.32 | 20.32 | 20.32 | 3.03 | 2.60 | 0.93 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 30.86 | 16.72 | 4.71 | 0.41 | 867.55 |
| 15.0 | 9.33 | 20.33 | 20.36 | 20.35 | 20.35 | 3.03 | 2.60 | 0.93 | 0.01 | 0.00 | 0.00 | 1.01 | 0.01 | 0.00 | 46.14 | 27.43 | 6.24 | 0.34 | 336.57 |
| 20.0 | 9.28 | 20.38 | 20.40 | 20.39 | 20.39 | 3.03 | 2.60 | 0.93 | 0.02 | 0.01 | 0.00 | 1.01 | 0.02 | 0.01 | 61.24 | 38.41 | 7.61 | 0.22 | 250.26 |

Initial height ($H_0$) = 19.25 *mm* Initial density ($\rho_f$) = 2.55 *g/cc* Initial diameter ($D_0$) = 20.30 *mm*
Weight in air ($w_a$) = 15.84 *g* Lubricant = $MoS_2$ Aspect ratio = 1.00

| Load *kN* | $H_f$ *mm* | $D_{CT}$ *mm* | $D_{CB}$ *mm* | $D_C$ *mm* | $D_B$ *mm* | $W_w$, *g* | $\rho_f$, *g/cc* | $\left(\frac{\rho_f}{\rho_{th}}\right)$ | $\varepsilon_z$ | $\varepsilon_\theta$ | $\varepsilon_{\theta'}$ | $e^{(\varepsilon_z-\varepsilon_\theta)}$ | $\psi$ | $\nu$ | $\sigma_z$ *MPa* | $\sigma_\theta$ *MPa* | $\sigma_m$ *MPa* | $n_i$ | $k_i$ *MPa* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7.0 | 19.18 | 20.33 | 20.34 | 20.33 | 20.33 | 6.21 | 2.55 | 0.91 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 21.56 | 11.30 | 3.42 | -- | -- |
| 15.0 | 19.03 | 20.35 | 20.38 | 20.36 | 20.37 | 6.20 | 2.55 | 0.91 | 0.01 | 0.00 | 0.00 | 1.01 | 0.00 | 0.00 | 46.06 | 26.12 | 6.65 | 0.42 | 736.07 |
| 20.0 | 18.70 | 20.48 | 20.53 | 20.50 | 20.55 | 6.20 | 2.55 | 0.91 | 0.03 | 0.02 | 0.01 | 1.02 | 0.33 | 0.13 | 60.57 | 39.99 | 6.86 | 0.30 | 172.88 |
| 25.0 | 18.17 | 20.69 | 20.76 | 20.73 | 20.80 | 6.17 | 2.57 | 0.92 | 0.06 | 0.04 | 0.02 | 1.04 | 0.33 | 0.30 | 74.10 | 50.75 | 7.78 | 0.29 | 171.80 |

Initial height ($H_0$) = 9.84 *mm* Initial density ($\rho_f$) = 2.51 *g/cc* Initial diameter ($D_0$) = 20.35 *mm*

Weight in air ($w_a$) = 7.97 $g$ Lubricant = $MoS_2$ Aspect ratio = 0.50

| Load $kN$ | $H_f$ $mm$ | $D_{CT}$ $mm$ | $D_{CB}$ $mm$ | $D_C$ $mm$ | $D_B$ $mm$ | $W_w$, $g$ | $\rho_f$, $g/cc$ | $\left(\frac{\rho_f}{\rho_{th}}\right)$ | $\varepsilon_z$ | $\varepsilon_\theta$ | $\varepsilon_{\theta'}$ | $e^{(\varepsilon_z-\varepsilon_\theta}$ | $\psi$ | $\nu$ | $\sigma_z$ $MPa$ | $\sigma_\theta$ $MPa$ | $\sigma_m$ $MPa$ | $n_i$ | $k_i$ $MPa$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5.0 | 9.81 | 20.38 | 20.38 | 20.38 | 20.38 | 3.18 | 2.51 | 0.89 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 15.33 | 2.17 | 4.39 | -- | -- |
| 10.0 | 9.77 | 20.38 | 20.39 | 20.39 | 20.39 | 3.18 | 2.51 | 0.89 | 0.01 | 0.00 | 0.00 | 1.01 | 0.00 | 0.00 | 30.63 | 6.31 | 8.11 | 0.45 | 1004.43 |
| 16.0 | 9.70 | 20.45 | 20.47 | 20.46 | 20.46 | 3.17 | 2.51 | 0.90 | 0.01 | 0.00 | 0.00 | 1.01 | 0.00 | 0.00 | 48.69 | 12.01 | 12.23 | 0.34 | 920.13 |
| 20.0 | 9.65 | 20.46 | 20.47 | 20.47 | 20.47 | 3.17 | 2.51 | 0.90 | 0.02 | 0.01 | 0.01 | 1.01 | 0.25 | 0.20 | 60.80 | 17.12 | 14.56 | 0.26 | 818.72 |

**Table 2 Effect of BPN-Generalizations tools**

| No. | Parameter | Early stopping | | Regularization | | Bayesian Network | |
|---|---|---|---|---|---|---|---|
| | | R | AARE% | R | AARE% | R | AARE% |
| 1 | $\varepsilon_z$ | 0.992 | 25.641 | 0.993 | 25.494 | 0.992 | 19.818 |
| 2 | $\varepsilon_\theta$ | 0.991 | 37.026 | 0.992 | 24.845 | 0.994 | 12.988 |
| 3 | $\varepsilon_{\theta'}$ | 0.993 | 20.700 | 0.990 | 22.570 | 0.991 | 18.314 |
| 4 | $S$ | 0.969 | 1.684 | 0.989 | 3.297 | 0.988 | 1.638 |
| 5 | $\psi$ | 0.934 | 12.5172 | 0.933 | 12.243 | 0.979 | 4.607 |
| 6 | $\nu$ | 0.946 | 16.292 | 0.935 | 16.948 | 0.974 | 7.430 |
| 7 | $\sigma_z$ | 0.994 | 4.153 | 0.978 | 15.023 | 0.993 | 3.641 |
| 8 | $\sigma_\theta$ | 0.991 | 9.325 | 0.966 | 15.134 | 0.991 | 5.053 |
| 9 | $\sigma_M$ | 0.948 | 10.786 | 0.912 | 19.522 | 0.913 | 13.198 |
| 10 | $n_i$ | 0.537 | 37.026 | 0.443 | 27.641 | 0.257 | 23.143 |
| 11 | $k_i$ | 0.423 | 31.772 | 0.212 | 22.340 | 0.124 | 15.371 |

**Table 3** Error comparison of NADP and hybrid model in MRBFNN

| NADP –MRBFNN Model with hidden layer size = Q(350 hidden nodes) | Hybrid model –MRBFNN with hidden layer size <Q (With k=200) |
|---|---|
| 0.993 | 0.991 |
| 0.993 | 0.992 |
| 0.989 | 0.988 |
| 0.991 | 0.988 |
| 0.988 | 0.989 |
| 0.977 | 0.979 |

| | |
|---|---|
| 0.992 | 0.991 |
| 0.991 | 0.989 |
| 0.884 | 0.893 |
| 0.902 | 0.878 |
| 0.876 | 0.866 |

**List of Figures:**



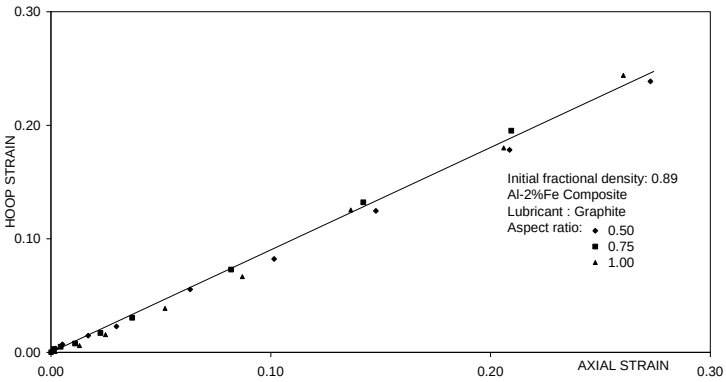


**Fig. 1(a)** Linear relationship

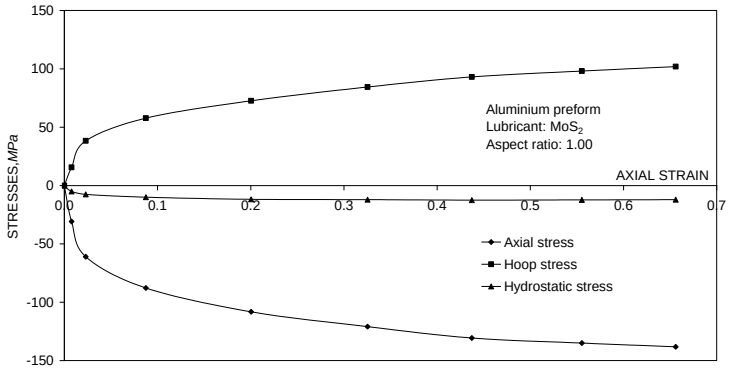


**Fig. 1(b)** Non-linear relationship

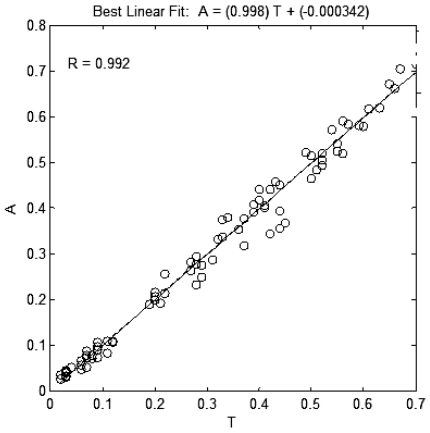


**Fig.2 (a)** Axial stress –Validation

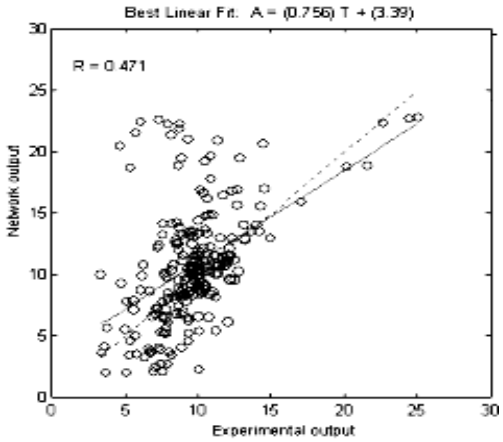


**Fig.2 (b**) Axial stress –Testing

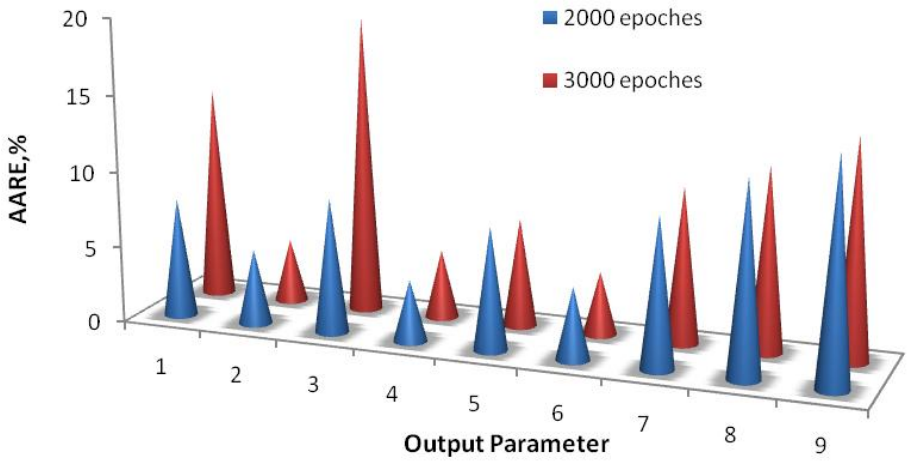


**Fig. 3** BPN with Non-Null Parameters

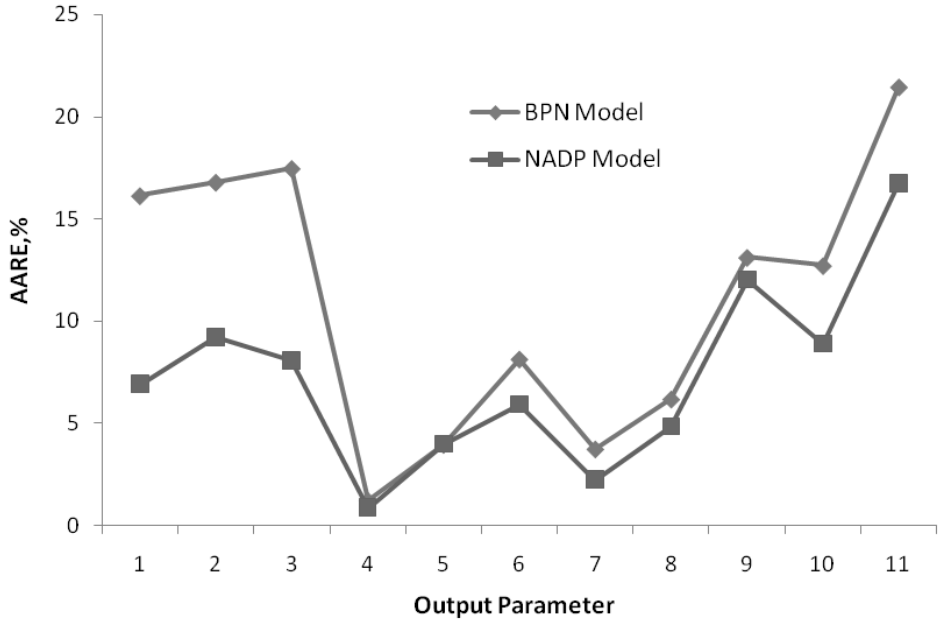


**Fig. 4** Comparison of BPN & MRBFNN-NADP model

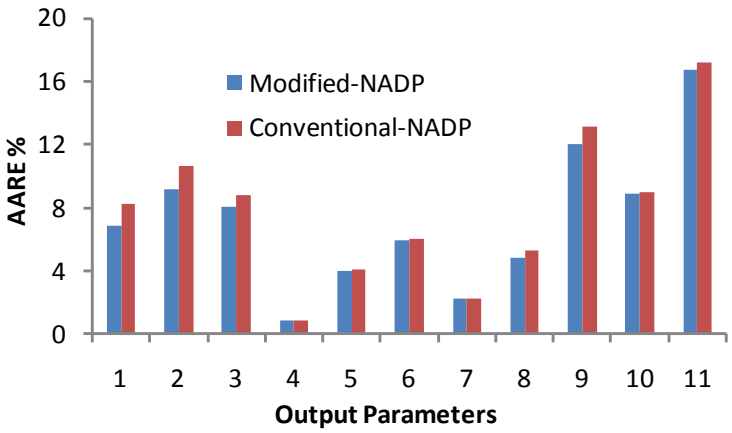


**Fig. 5** Comparison of Conventional RBF and MRBFNN

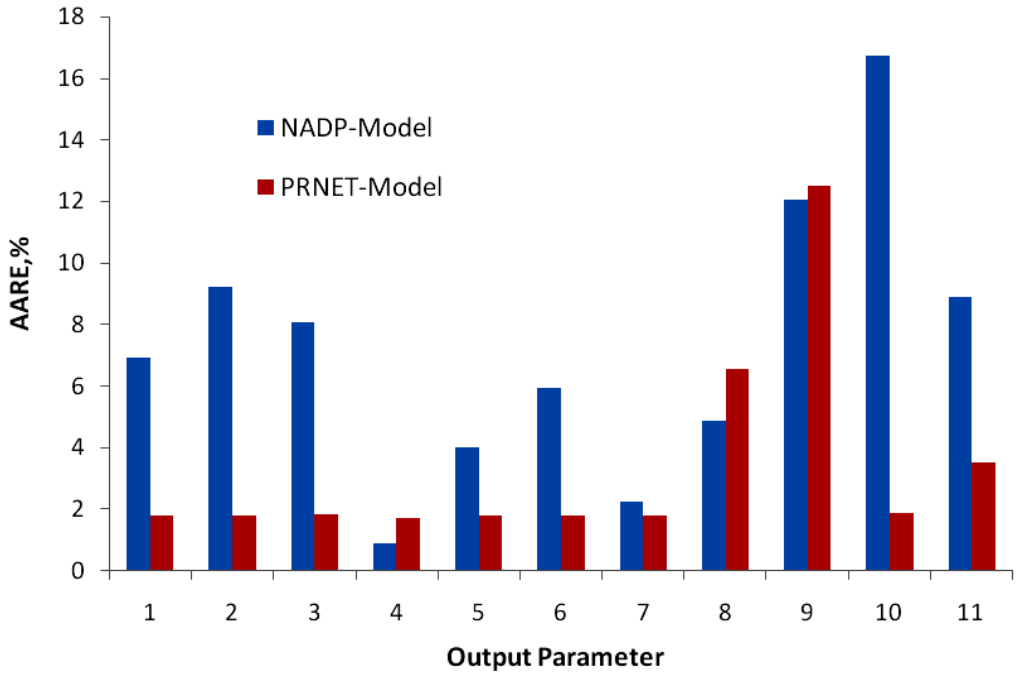


**Fig. 6** NADP model Vs. PRNET model