CHAPTER (7)

ARTIFICIAL NEURAL NETWORK

7.1 Introduction

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements **[153]**.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in the Figure 7.1. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this *supervised learning*, to train a network.

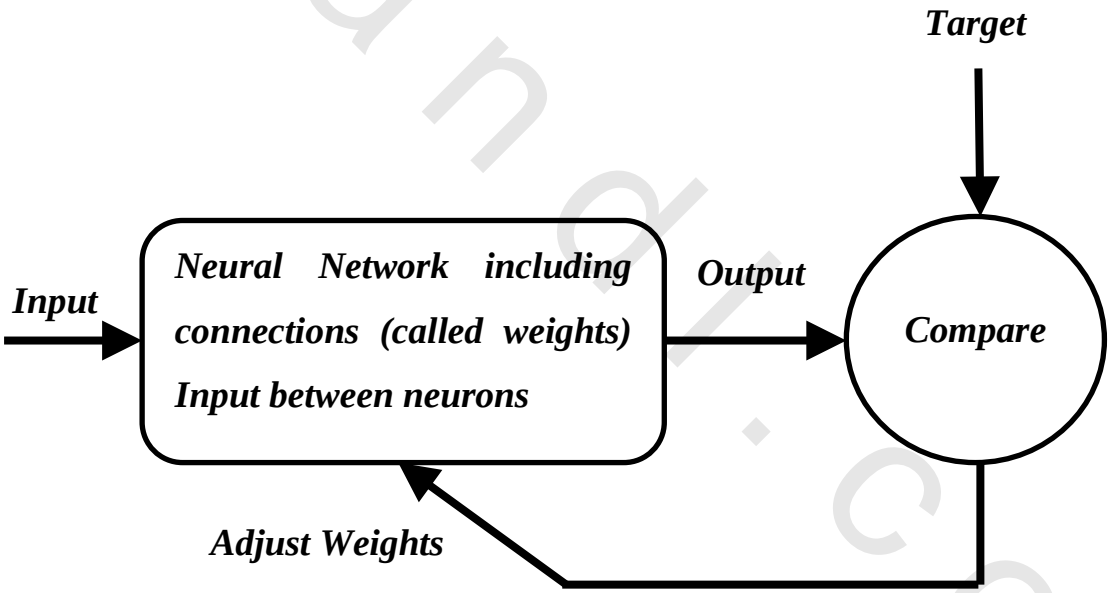


Figure 7.1.Schematic illustration of ANN project cycle.

Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as “on line” or “adaptive” training.

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

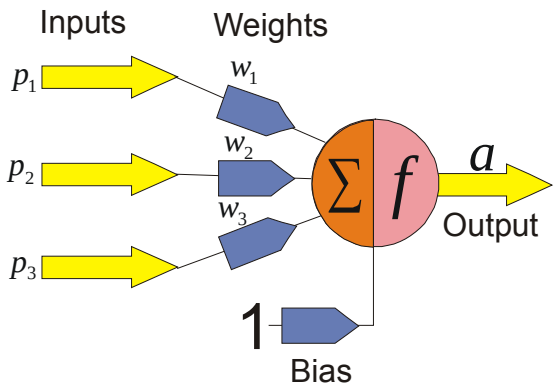
Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings. Throughout the toolbox emphasis is placed on neural network paradigms that build up to or are themselves used in engineering, financial and other practical applications.

The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed directly. In summary, there are a variety of kinds of design and learning techniques that enrich the choices that a user can make.

The field of neural networks has a history of some five decades but has found solid application only in the past fifteen years, and the field is still developing rapidly. Thus, it is distinctly different from the fields of control systems or optimization where the terminology, basic mathematics, and design procedures have been firmly established and applied for many years.

7.2 The Key Elements of Neural Networks

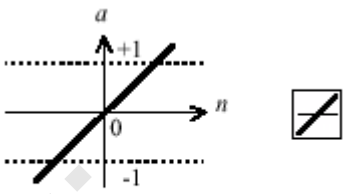
1. Neural computing requires a number of neurons, to be connected together into a neural network. Neurons are arranged in layers as shown in the Figure 7.2.



$$a = f(p_1w_1 + p_2w_2 + p_3w_3 + b) = f\left(\sum p_iw_i + b\right)$$

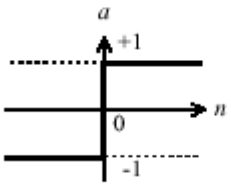
Figure 7.2.Schematic representation of Artificial Neural networks ANN.

2. Each neuron within the network is usually a simple processing unit which takes one or more inputs and produces an output. At each neuron, every input has an associated weight which modifies the strength of each input. The neuron simply adds together all the inputs and calculates an output to be passed on.
3. The activation function is generally non-linear. Linear functions are limited because the output is simply proportional to the input.



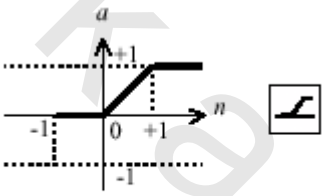
a = purelin(n)

Linear Transfer Function



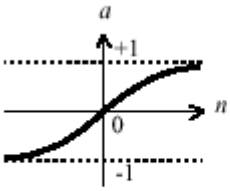
a = hardlims(n)

Symmetric Hard Limit Trans. Funct.



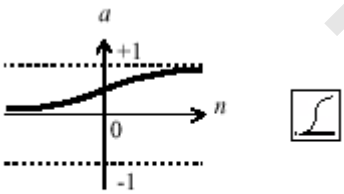
a = satlin(n)

Satlin Transfer Function



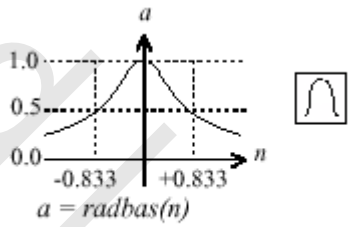
a = tansig(n)

Tan-Sigmoid Transfer Function



a = logsig(n)

Log-Sigmoid Transfer Function



a = radbas(n)

Radial Basis Function

7.3 Training methods

7.3.1 Supervised learning

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked. The set of data which enables the training is called the training set. During the training of a network the same set of data is processed many times as the connection weights are ever refined. Example architectures: Multilayer perceptrons.

7.3.2 Unsupervised learning

In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption.

7.4 Network Architectures

Two or more of the neurons shown earlier can be combined in a layer, and a particular network could contain one or more such layers. First consider a single layer of neurons.

7.4.1 Single Layer of Neurons

A one-layer network with *R* input elements and *S* neurons is presented in the Figure 7.3.

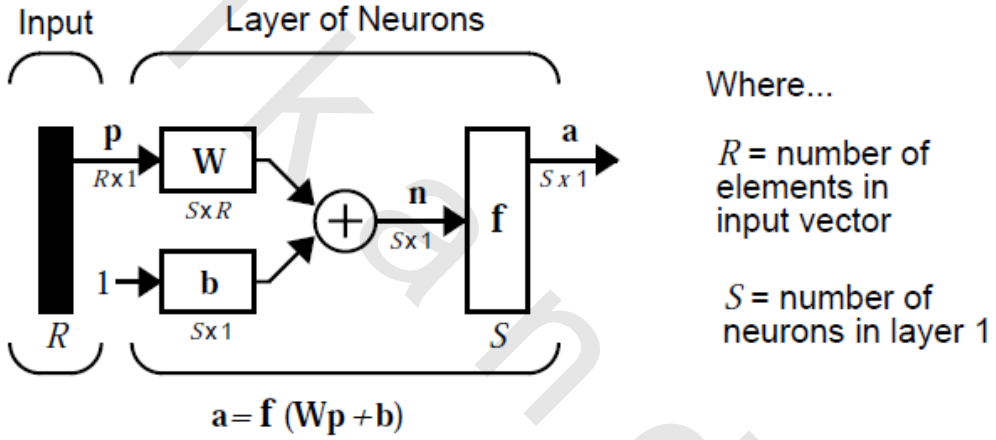


Figure 7.3.Schematic Illustration of Single Layer of Neurons **[153]**

Here p is an *R* length input vector, *W* is an $S \times R$ matrix, and *a* and *b* are *S* length vectors. As defined previously, the neuron layer includes the weight matrix, the multiplication operations, the bias vector *b*, the summer, and the transfer function boxes.

7.4.2 Multiple Layers of Neurons

A network can have several layers. Each layer has a weight matrix *W*, *a* bias vector *b*, and an output vector *a*. To distinguish between the weight matrices, output vectors, etc., for each of these layers in our figures, we append the number of the layer as a superscript to the variable of interest. You can see the use of this layer notation in the three-layer network shown below, and in the equations at the Figure 7.4.

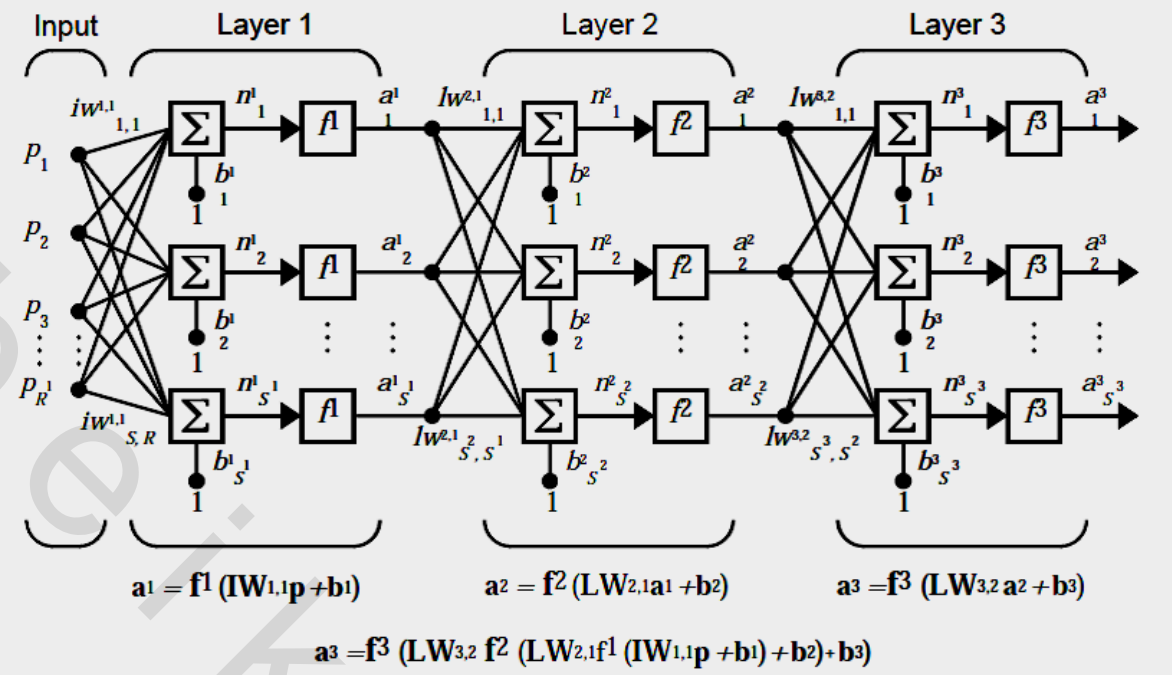


Figure 7.4.Schematic Illustration of Multi Layer of Neurons **[153]**

The network shown above has R^1 inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, etc. It is common for different layers to have different numbers of neurons. A constant input *1* is fed to the biases for each neuron.

7.5 Mean Square Error

Like the perceptron learning rule, the least mean square error *(LMS)* algorithm is an example of supervised training, in which the learning rule is provided with a set of examples of desired network behavior:

$$\{P_1, t_1\}, \{P_2, t_2\}, \ldots\ldots, \{P_Q, t_Q\}$$

Here P_q is an input to the network, and t_q is the corresponding target output. As each input is applied to the network, the network output is compared to the target. The error is calculated as the difference between the target output and the network output. We want to minimize the average of the sum of these errors.

$$mse = \frac{1}{Q}\sum_{k=1}^{Q} e(k)^2 = \frac{1}{Q}\sum_{k=1}^{Q} \left(t(k) - a(k)\right)^2$$

The *LMS* algorithm adjusts the weights and biases of the linear network so as to minimize this mean square error.

7.6 Neural Networks in the Field of fatigue failure of Composite Materials

Artificial neural networks (ANN) are one of the Artificial intelligence concepts that have proved to be useful for various engineering applications. As the study of fatigue failure of composite materials needs a large number of experiments as well as long time, so there is a need for new computational technique to expand the spectrum of the results and to save time.

A wide literature review showed that the fatigue behavior of composite materials is dependent on many factors. However, it is impossible to include all of them in a single laboratory test.

Modeling of these factors effects involves the development of a mathematical tool derived from experimental data. Once the model is established it can significantly reduce the experimental work involved in designing new polymer composite. For this reason, Artificial Neural Networks (ANN) has recently been introduced into the field of polymer composites **[154]**.

Artymiak et al. **[155]** used ANN to estimate the finite-life fatigue strength and fatigue limit of steel. Their predictions using ANN were found to be superior to those obtained with conventional methods for calculating the fatigue strength.

Venkatesh and Rack **[156]** trained a back propagation neural network to predict the elevated temperature creep-fatigue behavior of Ni-based alloy INCONEL 690. They concluded that the prediction accuracy using a few iterations and a simple network architecture showed significant improvement when compared to Con-Manson, linear life fraction and hysteresis energy prediction techniques.

Pleune and Chopra **[157]**also trained a back propagation network to predict the fatigue life of carbon and low alloy steels for specified sets of loading and environmental conditions They showed that ANN have great potential for predicting environmentally assisted corrosion due to the fact that the predictions are purely based on data and not on pre-conceptions. Another advantage was that the ANN could interpolate effects by learning trends and patterns when complete data sets are not available.

Elhadary **[86, 87]** suggested an artificial neural network (ANN) for woven-roving Glass Fiber-Reinforced Polyester (GFRP) with ($[\pm 45°]_{2s}$ and $[0,90°]_{2s}$) fiber orientations under combined bending and torsional moment, out/In-of-phase, with different fluctuating

stresses, and with different negative stress ratios. He used the feed-forward NN and a generalized regression NN to predict the number of cycles to failure (N), and he found that:

1. The used of feed-forward NN and generalized regression NN is suitable for life prediction for out/In-of-phase loading conditions for GFRP.
2. The used of feed-forward NN is more suitable and accurate than the generalized regression NN one of representing the fatigue behavior for composite materials.
3. A general FFNN was presented for GFRP which can predict the fatigue life at any stress ratio with any phase angle for certain fiber orientation.

Mohamed **[88]** Designed an artificial neural network (ANN) for woven-roving Glass Fiber-Reinforced Polyester (GFRP) with ($[30°,-60°]_{2s}$ $[\pm45°]_{2s}$, $[0,90°]_{2s}$ and $[-30°,60°]_{2s}$) fiber orientations under combined bending and torsional moment, with different negative and positive stress ratios. He used the feed-forward NN and a generalized regression NN to predict the number of cycles to failure (N), then he showed that the main goal of the Artificial Neural Networks (ANN) design is predicting non-experimental data not included in experimental test. He predicted the number of cycles to failure (N) for stress ratios -0.15, -0.65 and -0.85 and random values of maximum normal stresses for all fiber orientation. He found that the present Artificial Neural Network is suitable and useful in predicting non-experimental data.

Lee et al. **[158]** carried out an ANN predication on the fatigue life of some carbon/glass fiber reinforced plastic (CFRP/GFRP) laminates. Three fatigue parameters (peak stress, minimum stress and probability of failure) as well as four mono-tonic mechanical properties (tensile strength, compression strength, tensile failure strain and tensile modulus) were selected as the ANN inputs, which were applied to predict the fatigue life of the composite as the output. They concluded that the ANN's can be trained at least to model constant – stress fatigue behavior as well as other current life predica-tion methods and can provide accurate representations of the stress, stress ratio and median life for carbon fiber composites from a quite small experimental dataset.

The ANN predictive results of Aymerich and Serra **[159]** confirmed that the properties of the basic element (lamina) and their orientation within the laminate strongly affect the fatigue performance of composite laminates. It is concluded that an ANN is a very attractive approach to predict fatigue life of laminate composites, although a larger data-set is needed when increasing the number of laminate parameters.

Al-Assaf and El-Kadi **[160]** applied the ANN approach to predict the fatigue life of unidirectional glass fiber/epoxy composite. Only the stress ratio (R = minimum stress/ maximum stress), the maximum stress and the fiber orientation angle were used as the ANN input, and the output was the number of cycles to fatigue failure. In order to improve the predictive accuracy, other types of ANN's were considered in a later work **[161]**.

El-Kadi and Al-Assaf **[161]** used different neural network structures to predict the fatigue life and compared the results with those previously obtained using FNN. As before, the fiber orientation angle, the maximum stress and stress ratio were the inputs to the network that will predict the fatigue life of the unidirectional glass fiber/epoxy composites. Four ANN architectures were investigated: modular neural network (MNN), Radial basis function network (RBF), self organizing features maps (SOFM) and Principal component analysis (PCA). They concluded that MNN with five sub-networks gave the best results. The normalized mean square-error was reduced from 14.27% in the case of FNN to 5.7% for MNN.

Choi et al. **[162]** studied the fatigue damage predication in notched composite laminates using an artificial network. The ANN model was developed to describe the split growth in notched AS4/3501-6 graphite/ epoxy quasi- isotropic laminates under con- stant amplitude fatigue. The ANN model is found to work well.

In the present study, a new technique to simulate and predict the fatigue life of woven-roving glass fiber reinforcement epoxy (GFRE) subjected to combined completely reversed bending moments and internal hydrostatic pressure with different pressure ratios ($P_r = 0, 0.25, 0.5, 0.75$) with two fiber orientations (θ), $[0,90°]_{3s}$ and $[\pm 45°]_{3s}$, with two methods of manufacturing M_1 and M_2 for each fiber orientation is presented.

7.7 Neural network to study the effect of Pressure ratio

An ANN is designed to predict the fatigue life of glass fiber reinforcement epoxy (GFRE) with the maximum bending stress σ_{max} and pressure ratio P_r, as the input and the number of cycles to failure N as the output. Three neural network structures, feed forward FFNN, generalized regression GRNN and radial basis RBNN are designed, trained and tested.

7.7.1 A feed-forward Neural Network, FFNN

A feed-forward NN structure is applied for training the data of $[0,90°]_{3s}$ and $[\pm 45°]_{3s}$ fiber orientations with two methods of manufacturing M_1 and M_2 for each fiber orientation

at different pressure ratios (P_r). Figure 7.4 shows the training performance of suggested feed-forward NN. Figures 7.5 to 7.8 represent the comparison between the experimental data and the feed forward neural network *FFNN* predicted number of cycles to failure at $P_r = 0, 0.5, 0.75$ for both fiber orientations with two methods of manufacturing. The results of this case show much satisfactory predication quality for this case study. Figures 7.9 to 7.12 shows the comparison between the experimental data and the feed forward neural network *FFNN* expected data at $P_r = 0.25$. From these Figures, it noted that the expected data from the suggested feed forward neural network *FFNN* are applicable with the experimental data, Tables 7.1 and 7.2.

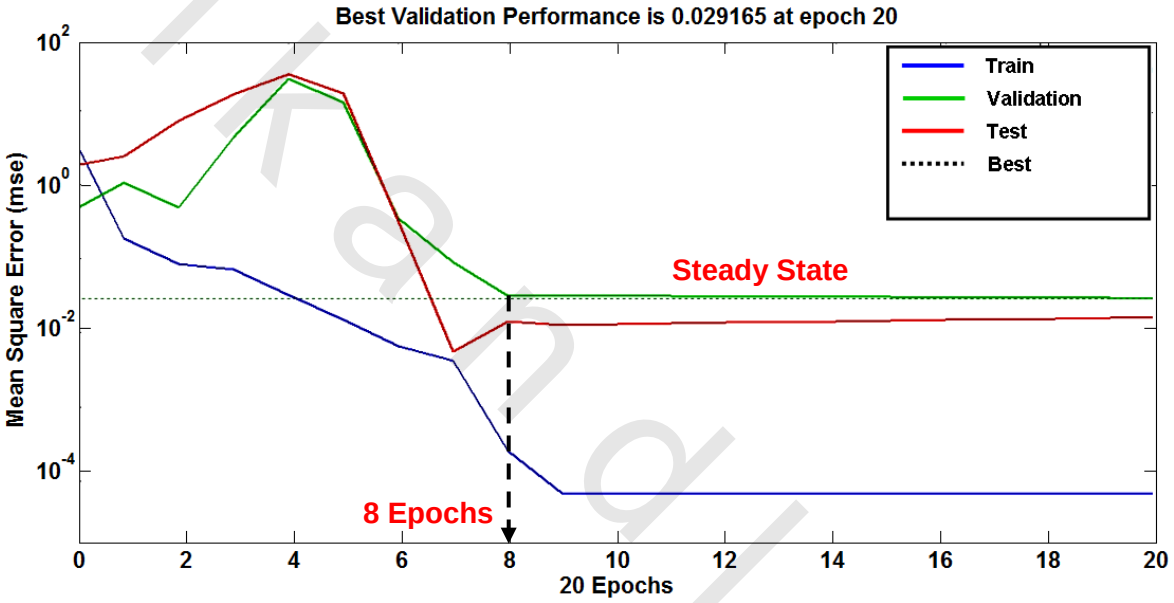


Figure 7.4 training performance of suggested feed-forward NN

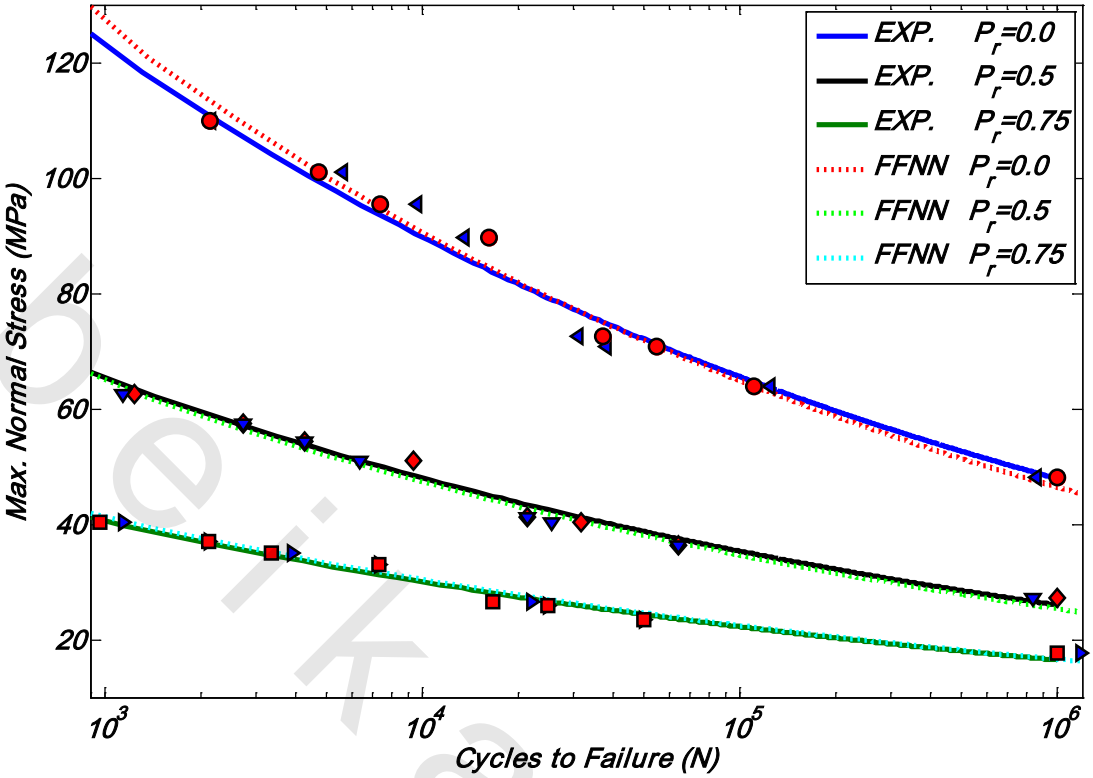


Figure 7.5 Comparison between the experimental data and the feed forward neural network FFNN predicted data for M_1, $[0,90°]_{3s}$ specimens with$P_r = 0, 0.5, 0.75$

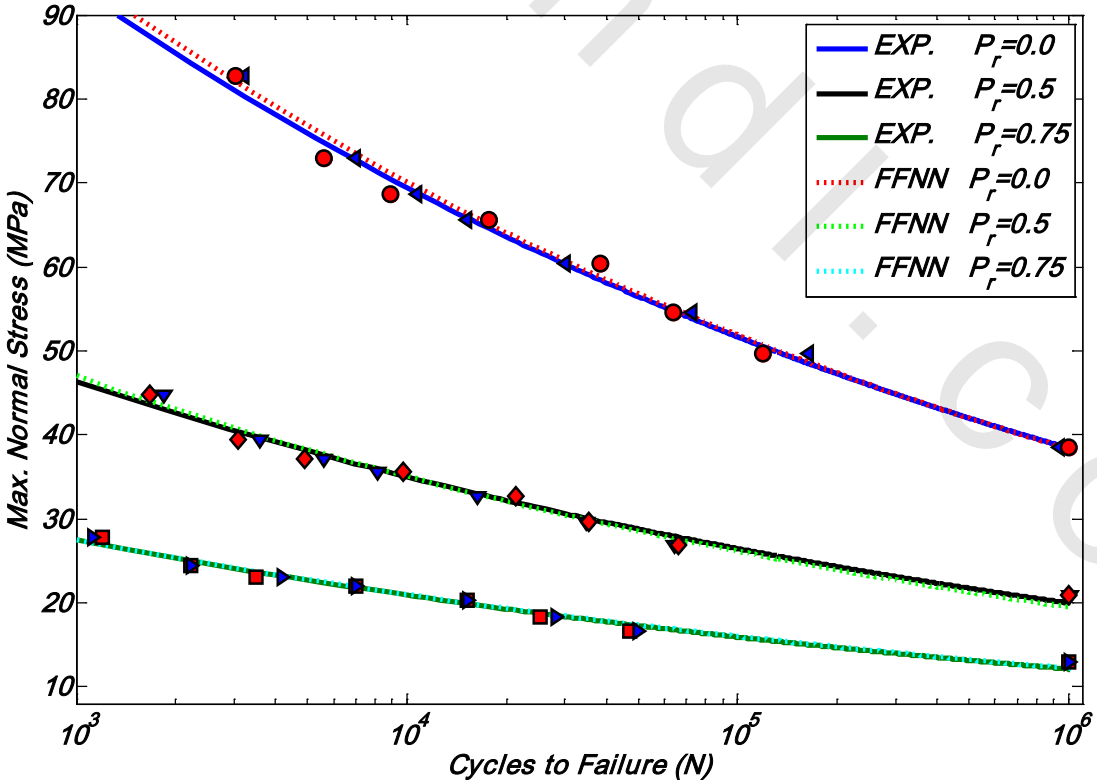


Figure 7.6 Comparison between the experimental data and the feed forward neural network FFNN predicted data for M_1, $[\pm 45°]_{3s}$ specimens with$P_r = 0, 0.5, 0.75$

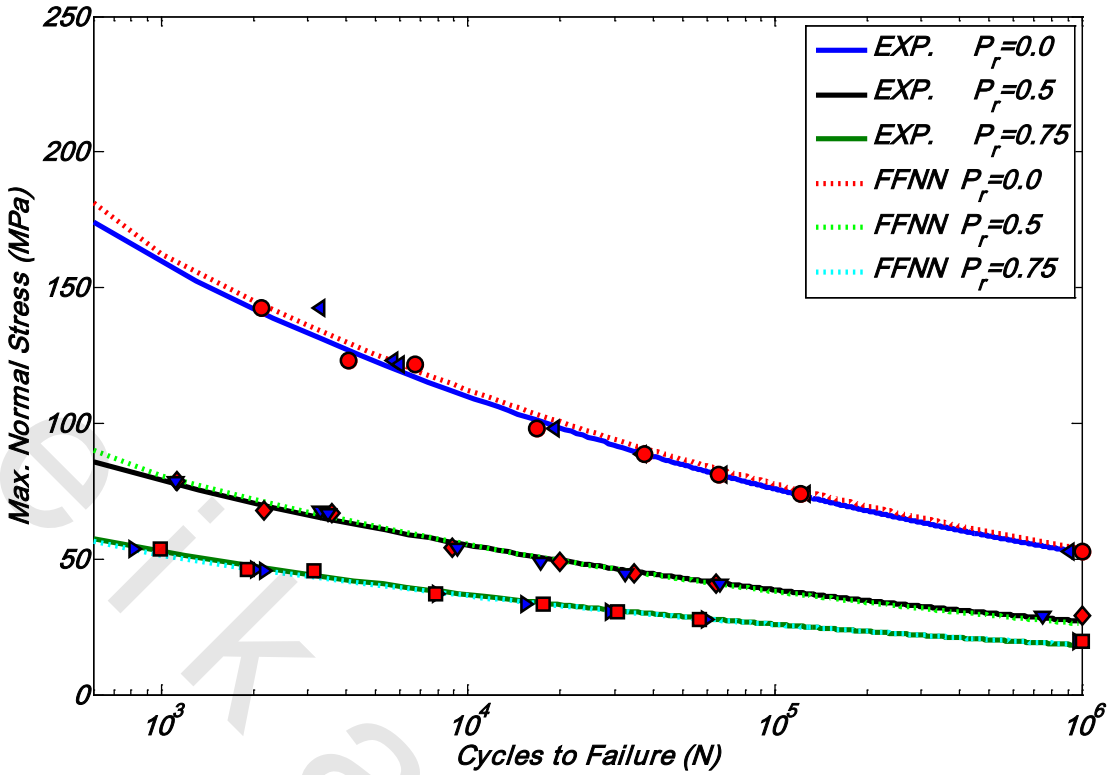


Figure 7.7 Comparison between the experimental data and the feed forward neural network FFNN predicted data for M_2, $[0,90°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

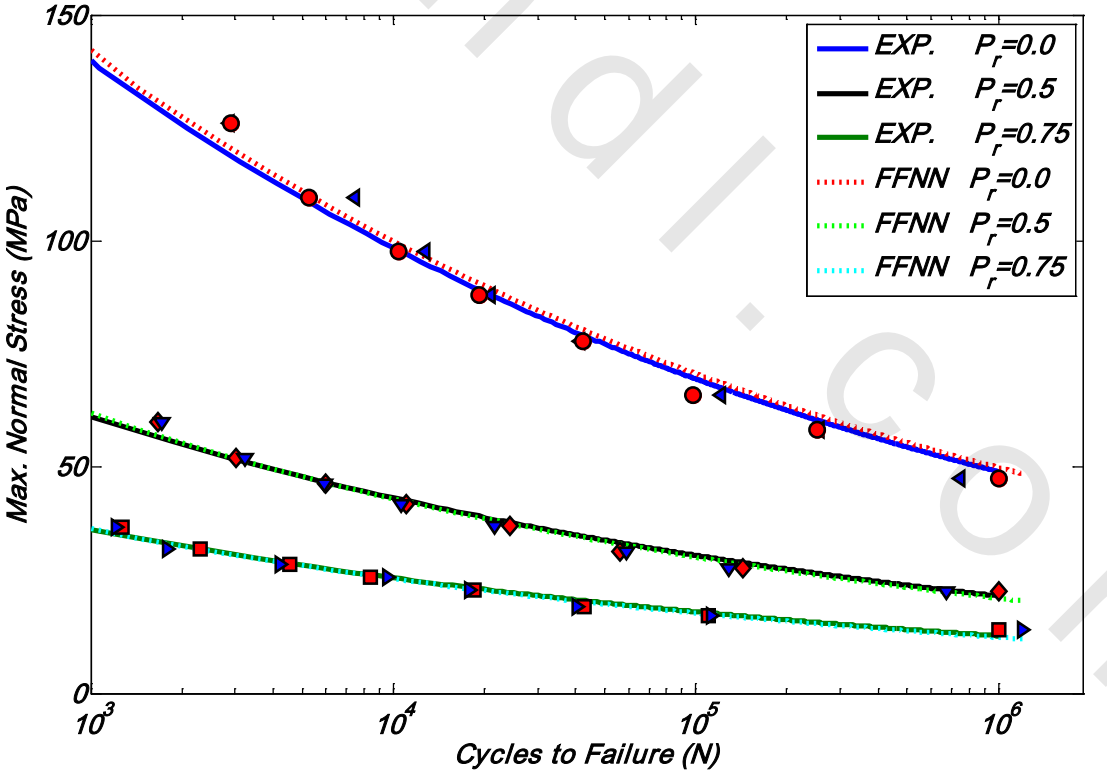


Figure 7.8 Comparison between the experimental data and the feed forward neural network FFNN predicted data for M_2, $[\pm 45°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

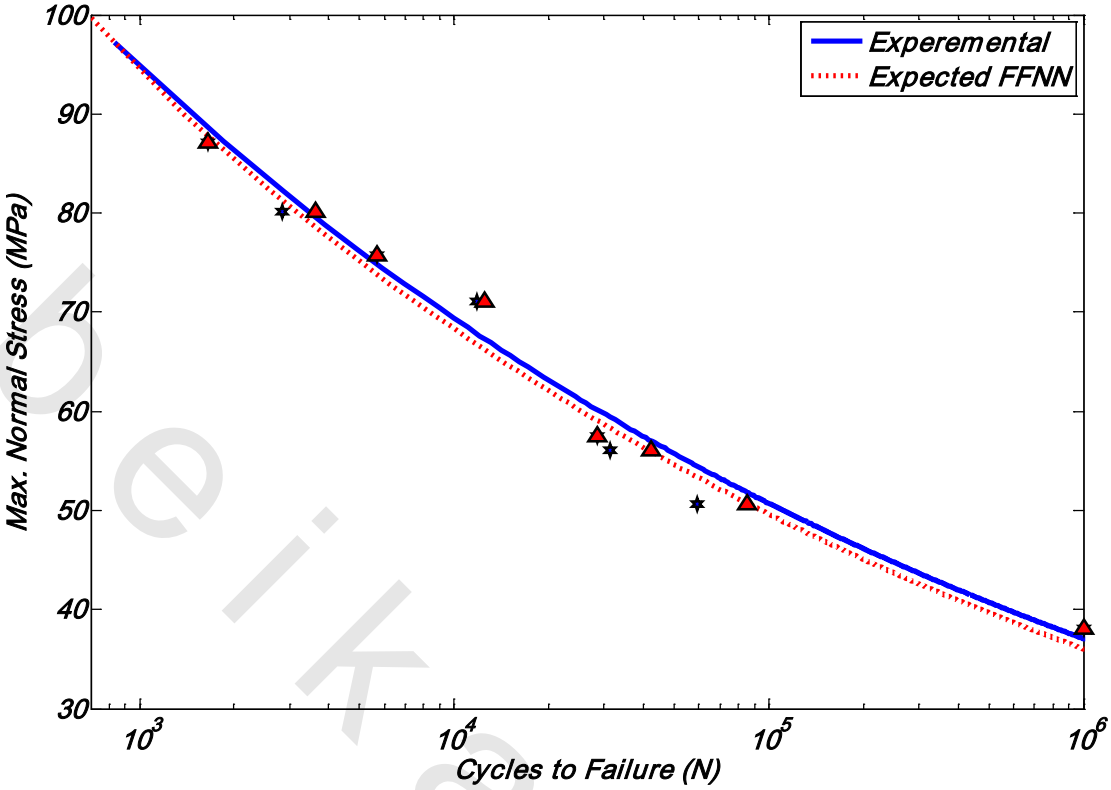


Figure 7.9 Comparison between the experimental data and the feed forward neural network FFNN Expected data for M_1, $[0,90°]_{3s}$ specimens with$P_r = 0.25$

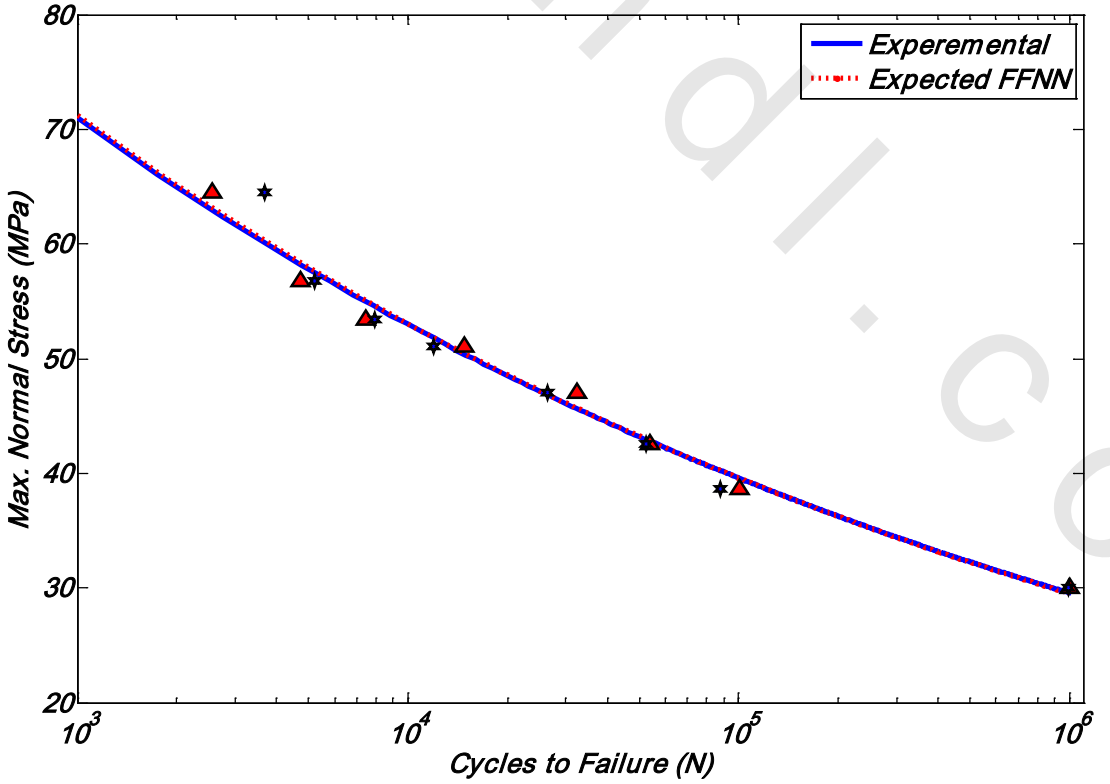


Figure 7.10 Comparison between the experimental data and the feed forward neural network FFNN Expected data for M_1, $[\pm 45°]_{3s}$ specimens with$P_r = 0.25$

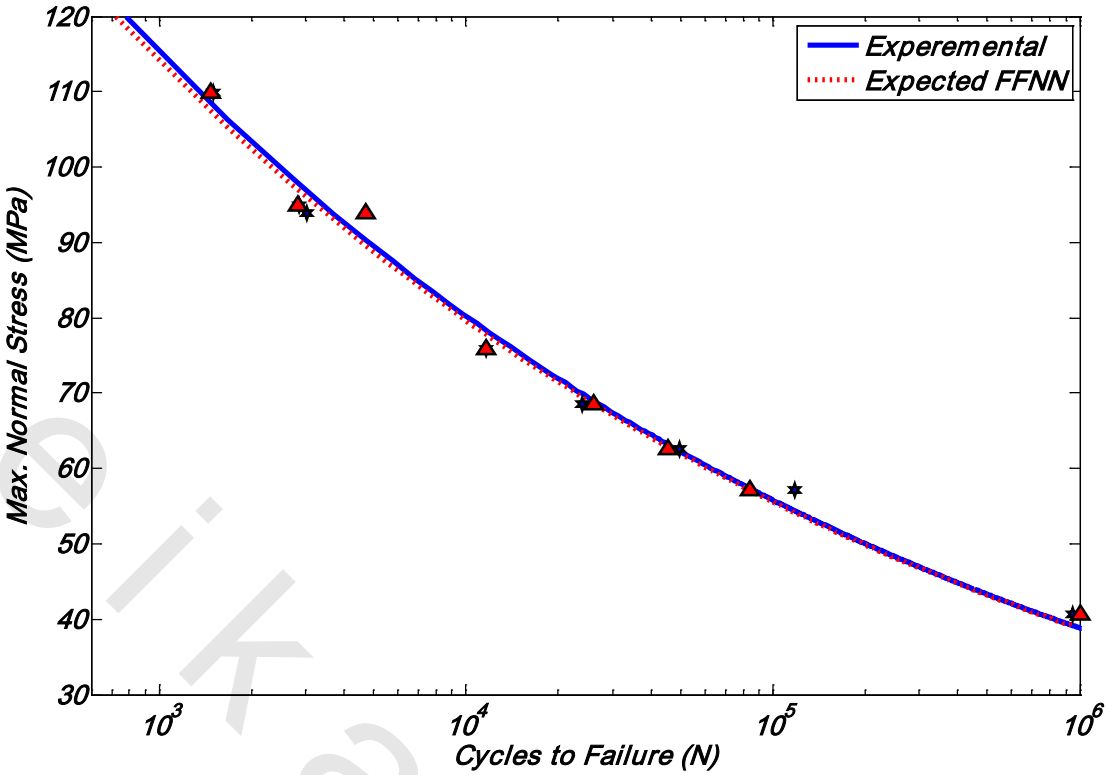


Figure 7.11 Comparison between the experimental data and the feed forward neural network FFNN Expected data for M_2, $[0,90°]_{3s}$ specimens with$P_r = 0.25$

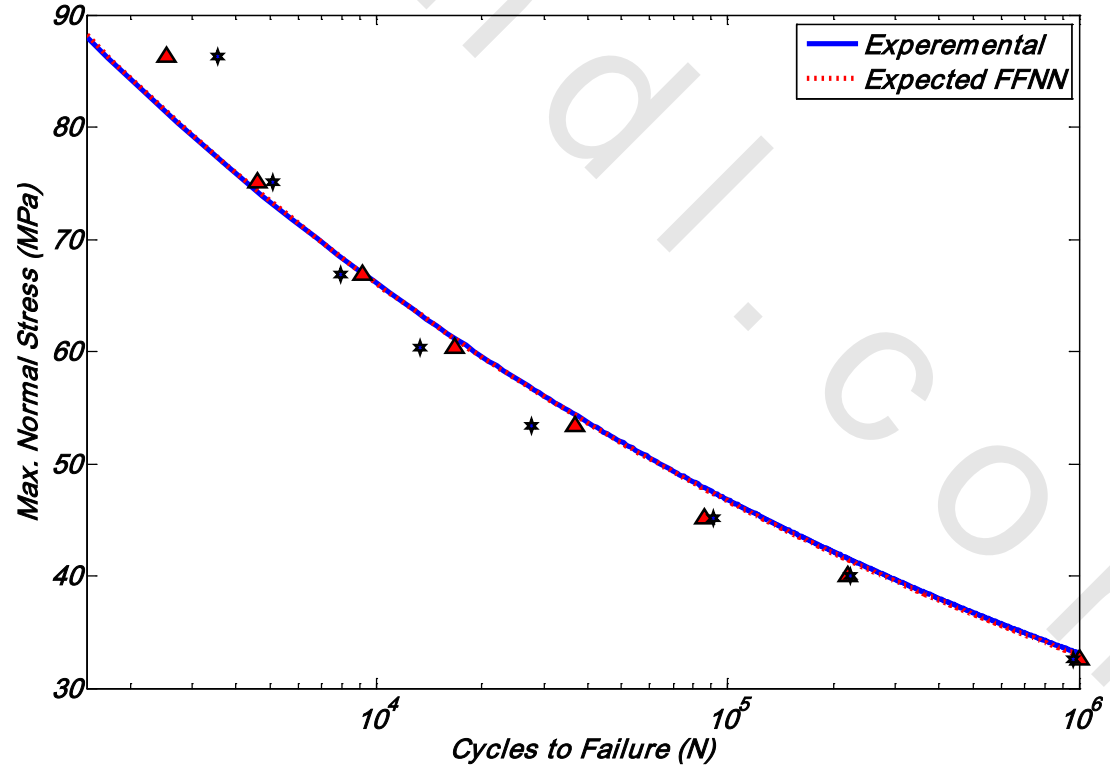


Figure 7.12 Comparison between the experimental data and the feed forward neural network FFNN Expected data for M_2, $[\pm 45°]_{3s}$ specimens with$P_r = 0.25$

7.7.2 Generalized regression Neural Network, GRNN

A generalized regression NN is designed based on two layers, the first, input layer, has a radial basis neurons while the second layer has purelinear ones **[153]**. This network is trained by the measured values of σ_{max}, Pr, θ. Figure 7.13 shows the Schematic Illustration of *GRNN* design for present study with input data σ_{max} , P_r , θ and N.

Figures 7.14 to 7.17 represent the comparison between the experimental data and the generalized regression neural network *GRNN* predicted number of cycles to failure at $P_r = 0, 0.5, 0.75$ for both fiber orientation and two methods of manufacturing. The results show much better predication quality for the case of $[\pm 45°]_{3s}$ than $[0,90°]_{3s}$ and, which represents the same conclusions found when using feed-forward NN discussed earlier, Tables 7.1 and 7.2.

Figures 7.18 to 7.21 shows the comparison between the experimental data and the generalized regression neural network *GRNN* expected data at $P_r = 0.25$.

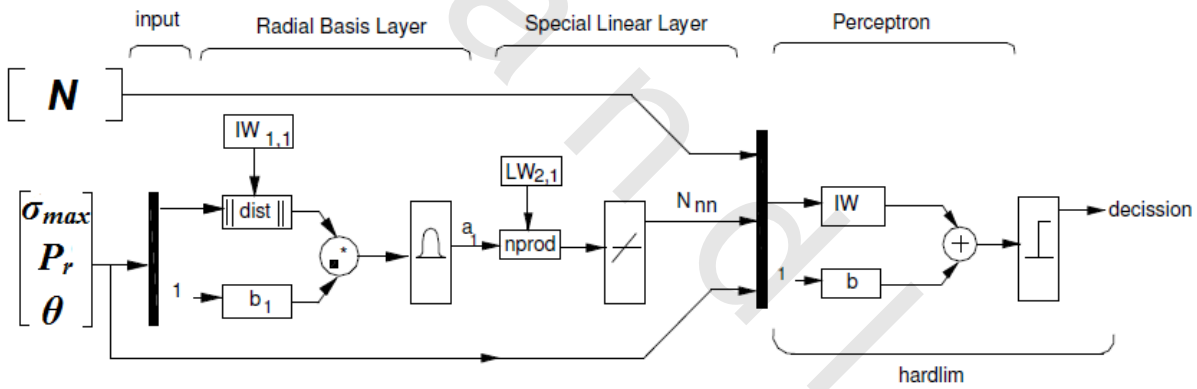


Figure 7.13 Schematic Illustration of GRNN design for present study with input data σ_{max} , P_r , θ and N **[154]**

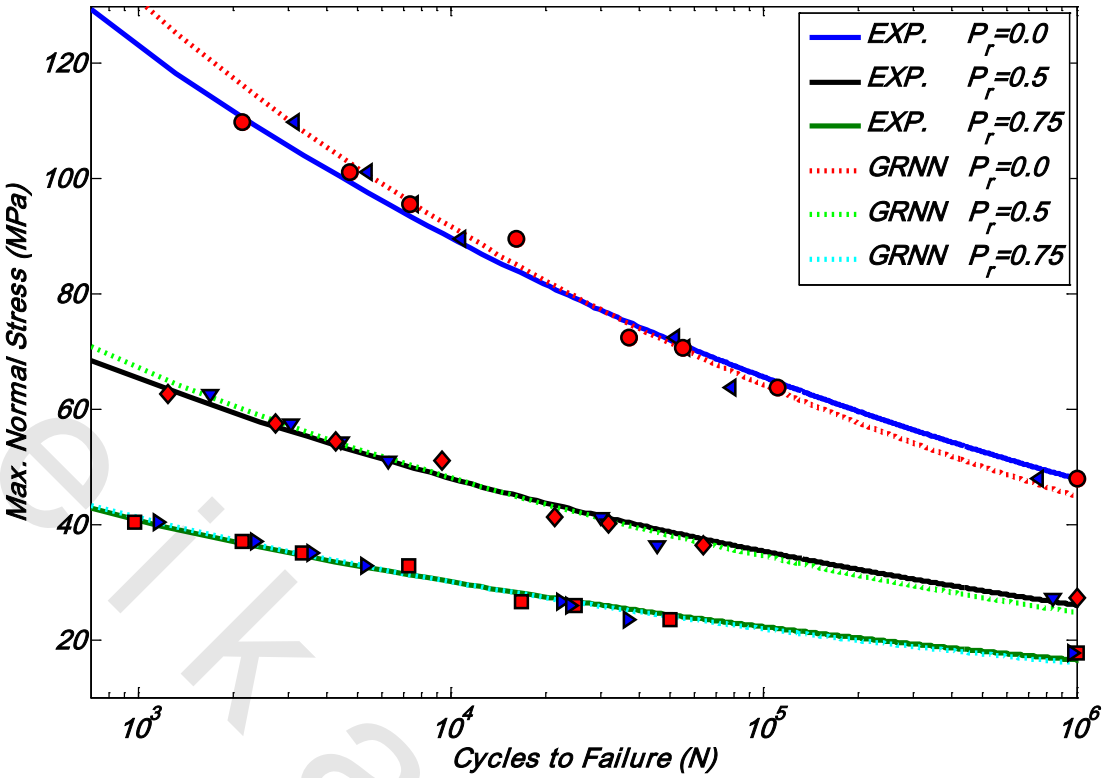


Figure 7.14 Comparison between the experimental data and the generalized regression neural network GRNN predicted data for M_1, $[0,90°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

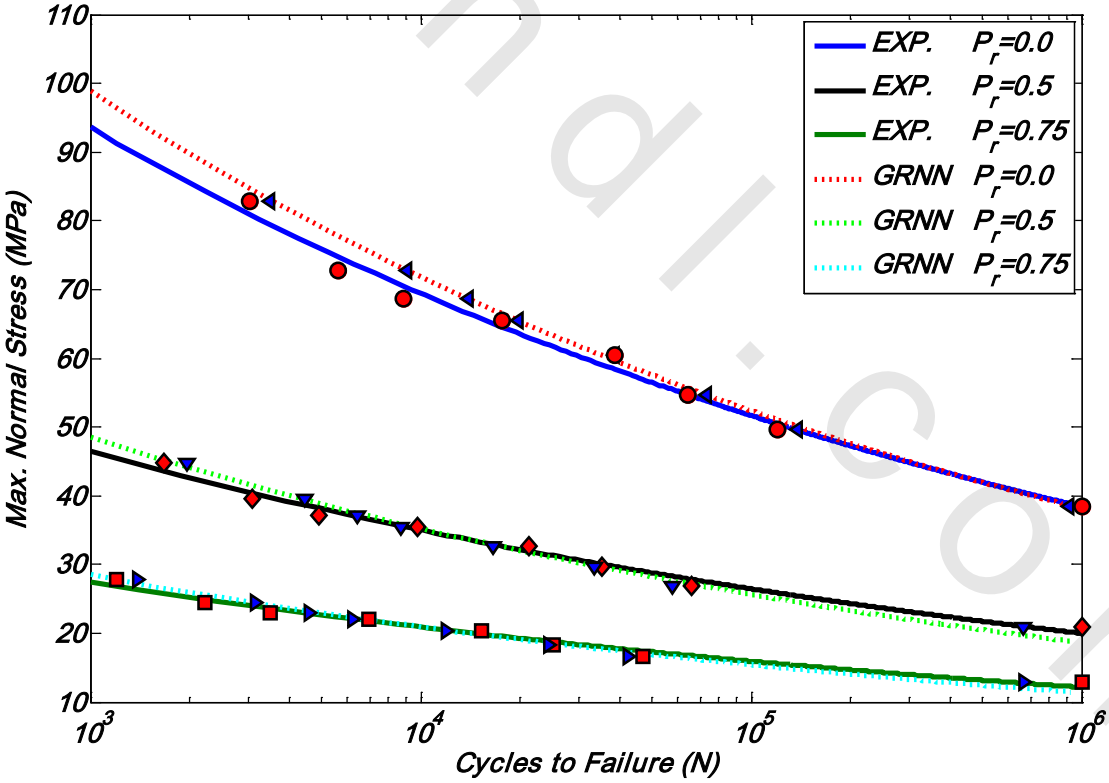


Figure 7.15 Comparison between the experimental data and the generalized regression neural network GRNN predicted data for M_1, $[\pm 45°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

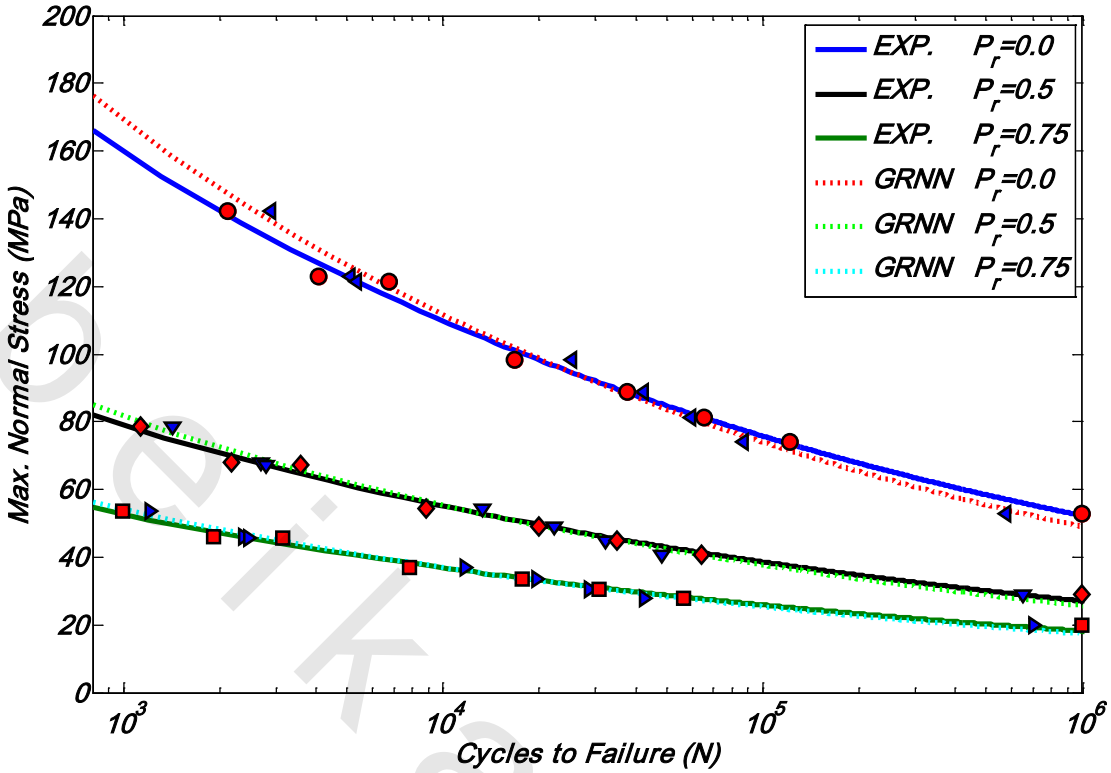


Figure 7.16 Comparison between the experimental data and the generalized regression neural network GRNN predicted data for M_2, $[0,90°]_{3s}$ specimens with$P_r = 0, 0.5, 0.75$

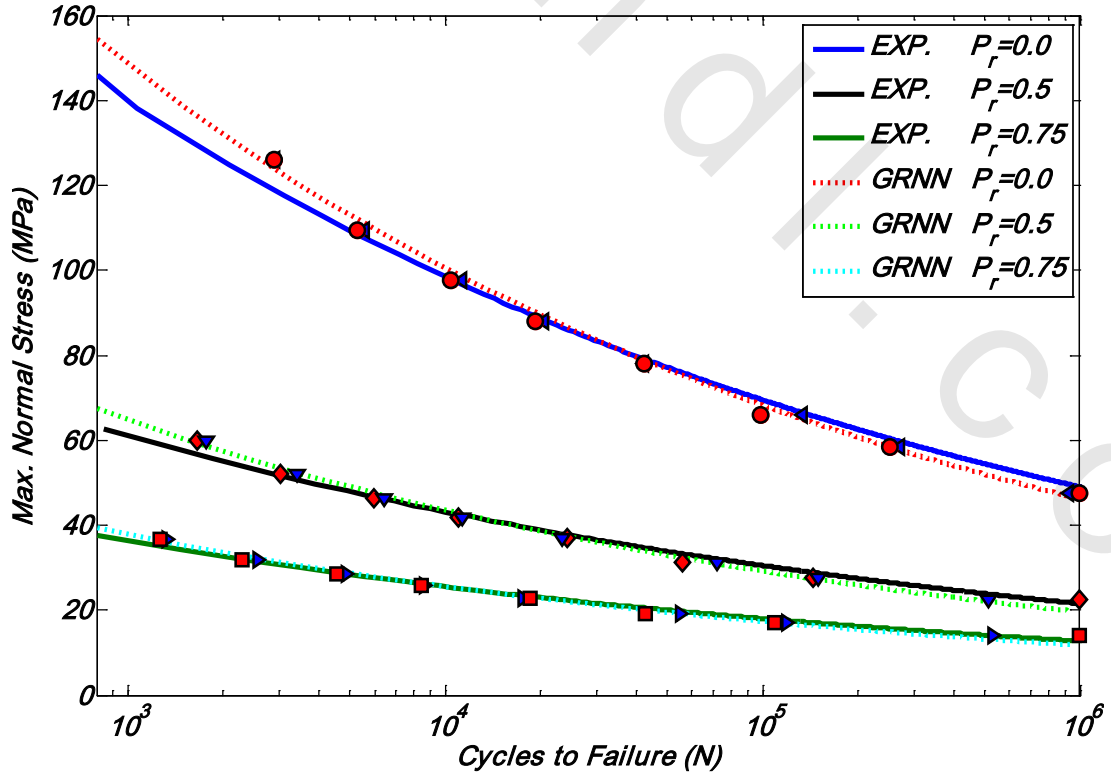


Figure 7.17 Comparison between the experimental data and the generalized regression neural network GRNN predicted data for M_2, $[\pm 45°]_{3s}$ specimens with$P_r = 0, 0.5, 0.75$

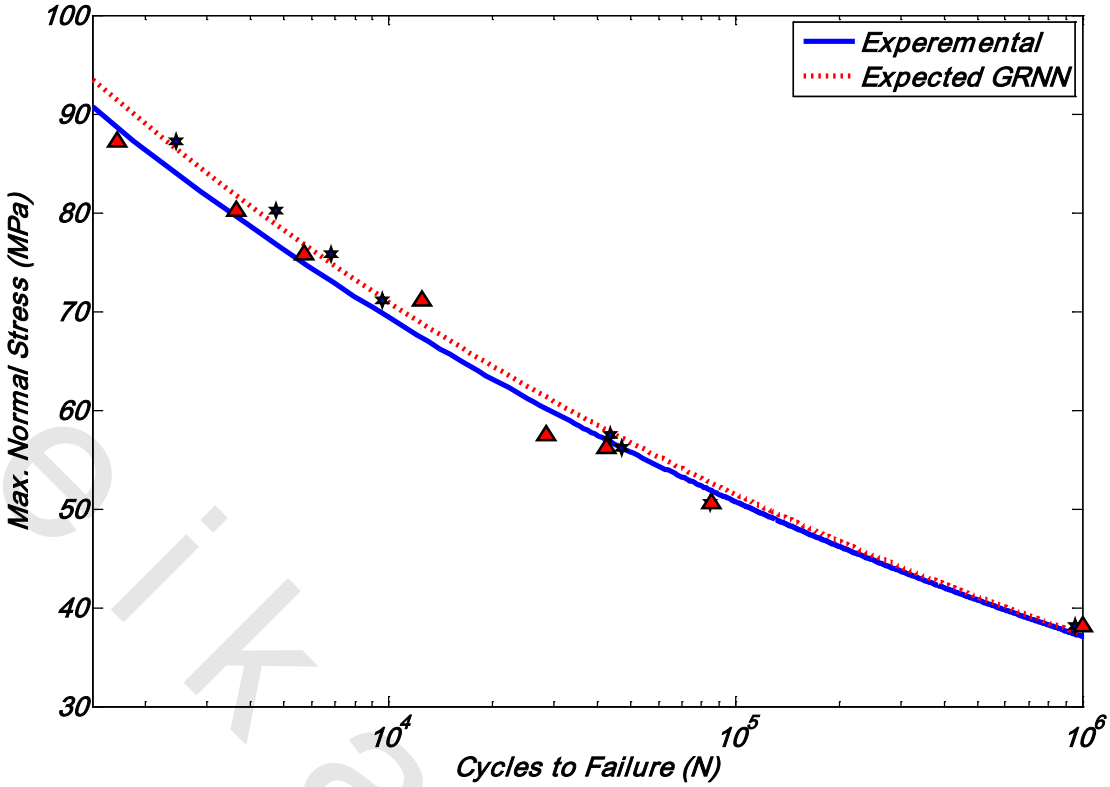


Figure 7.18 Comparison between the experimental data and the generalized regression neural network GRNN Expected data for M_1, $[0,90°]_{3s}$ specimens with $P_r = 0.25$

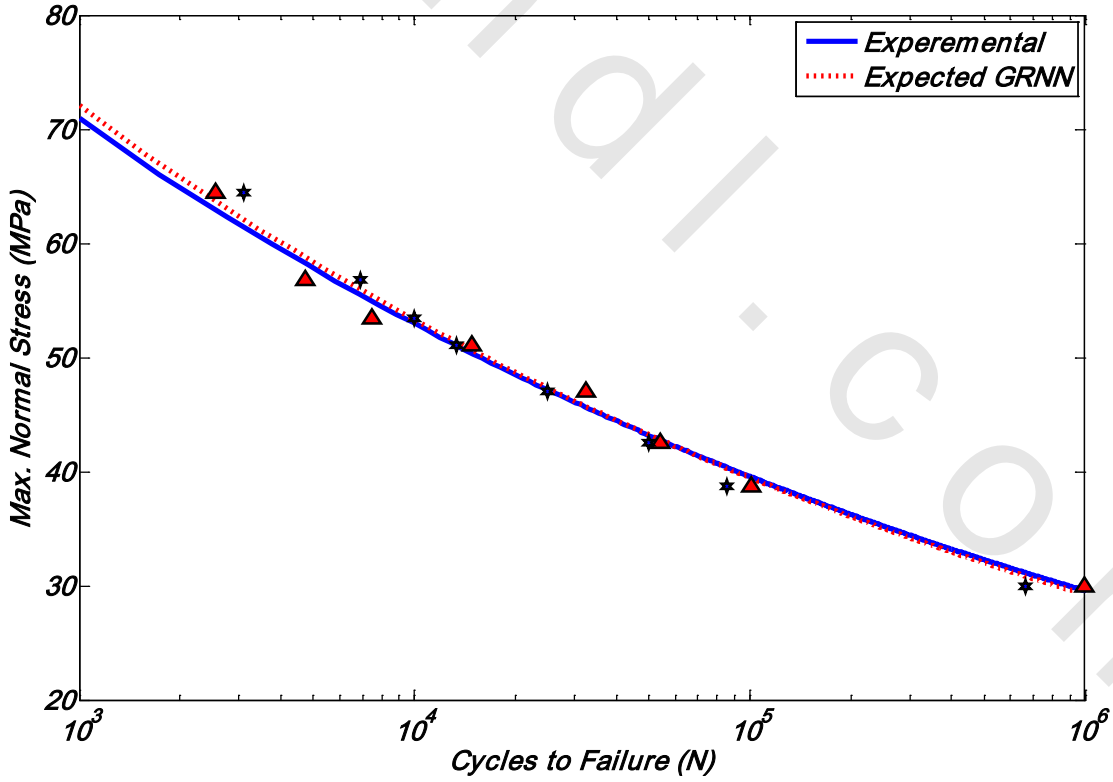


Figure 7.19 Comparison between the experimental data and the generalized regression neural network GRNN Expected data for M_1, $[\pm 45°]_{3s}$ specimens with $P_r = 0.25$

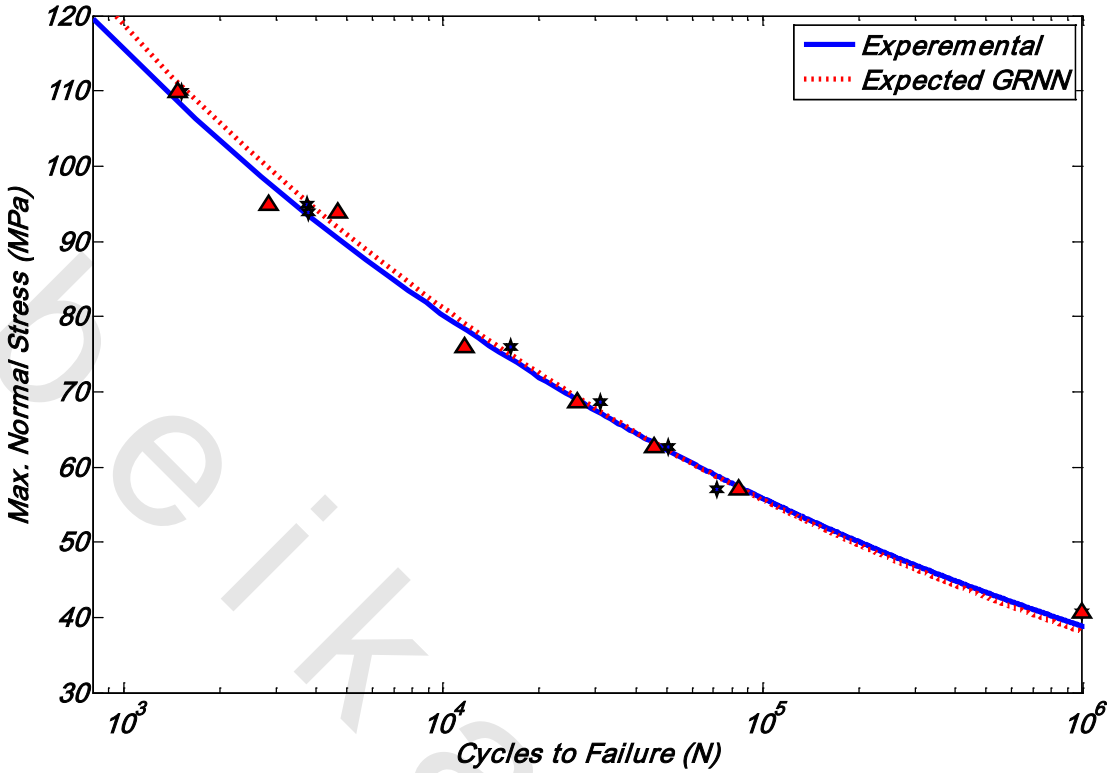


Figure 7.20 Comparison between the experimental data and the generalized regression neural network GRNN Expected data for M_2, $[0,90°]_{3s}$ specimens with $P_r = 0.25$

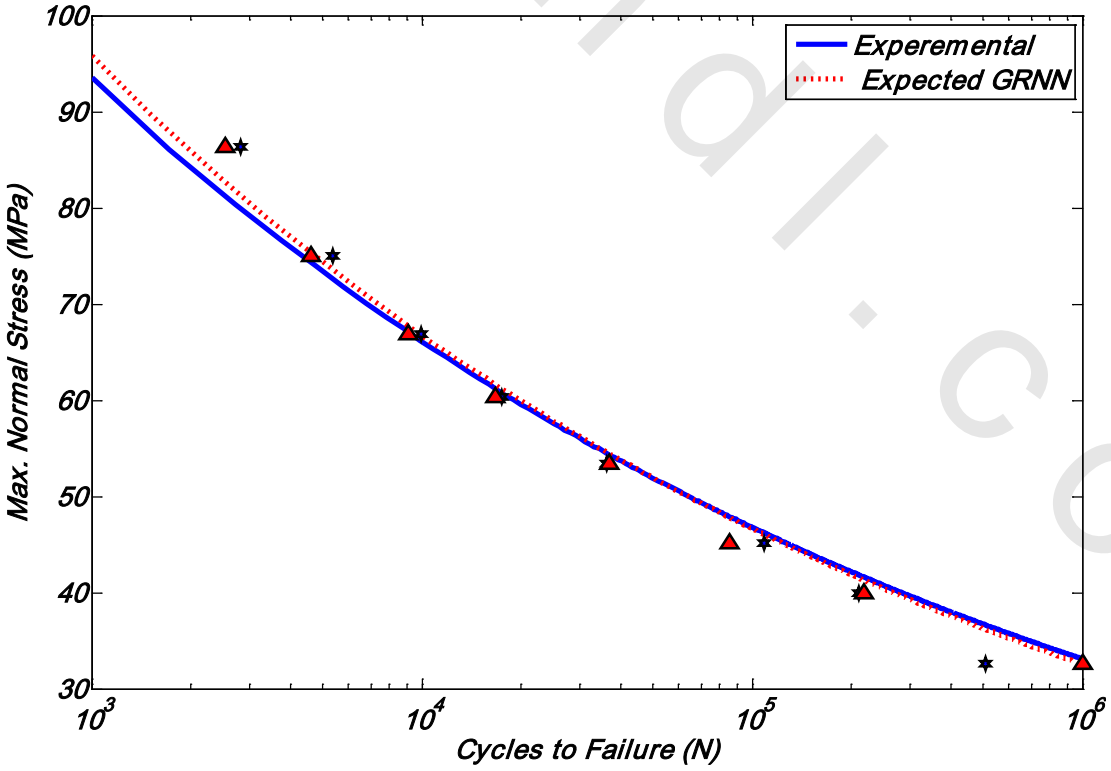


Figure 7.21 Comparison between the experimental data and the generalized regression neural network GRNN Expected data for M_2, $[\pm 45°]_{3s}$ specimens with $P_r = 0.25$

7.7.3 Radial Basis Neural Network, RBNN

Radial basis networks can require more neurons than standard feed forward back propagation networks, but often they can be designed in a fraction of the time it takes to train standard feed forward networks. They work best when many training vectors are available **[163]**.

Radial basis networks consist of two layers, a hidden radial basis layer and an output linear layer. This network is trained by the measured values of σ_{max}, P_r, θ.

Figures 7.22 to 7.25 represent the comparison between the experimental data and the Radial basis neural network RBNN predicted number of cycles to failure at $P_r = 0, 0.5, 0.75$ for both fiber orientation and two methods of manufacturing. Figures 7.26 to 7.29 shows the comparison between the experimental data and the Radial basis neural network *RBNN* expected data at $P_r = 0.25$.

From these Figures, it noted that the expected data from the suggested basis neural network RBNN are suitable with the experimental data, Tables 7.1 and 7.2.

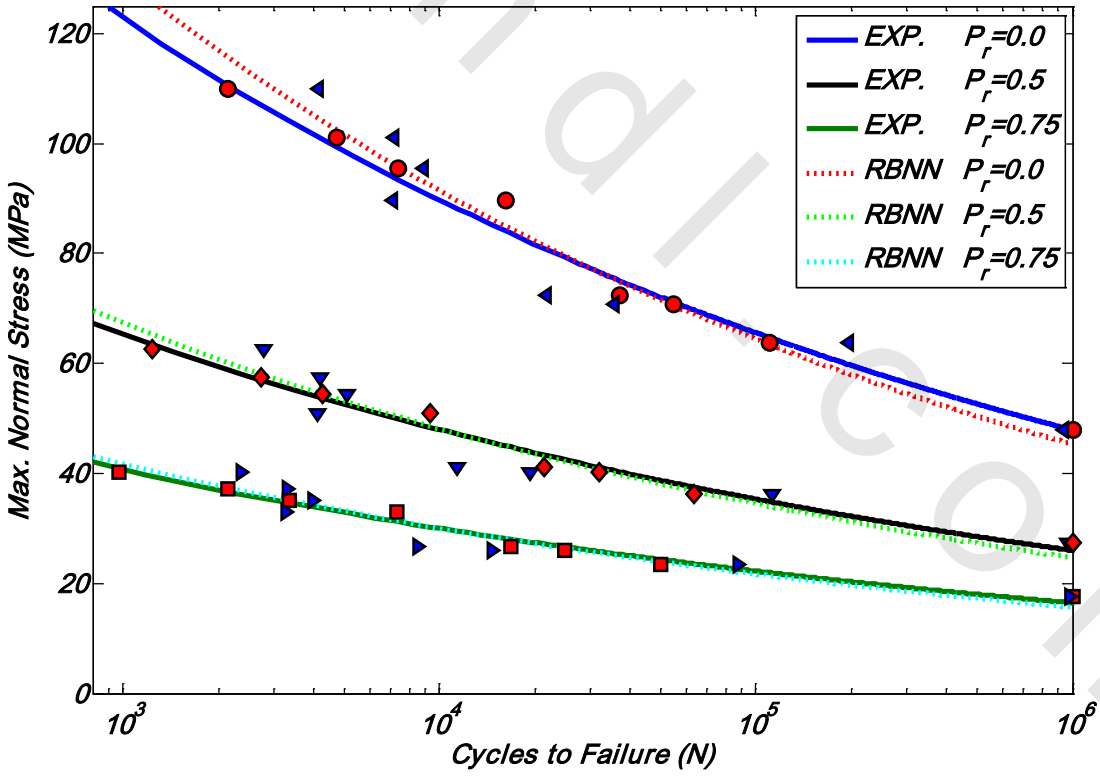


Figure 7.22 Comparison between the experimental data and the Radial basis neural network RBNN predicted data for M_1, $[0,90°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

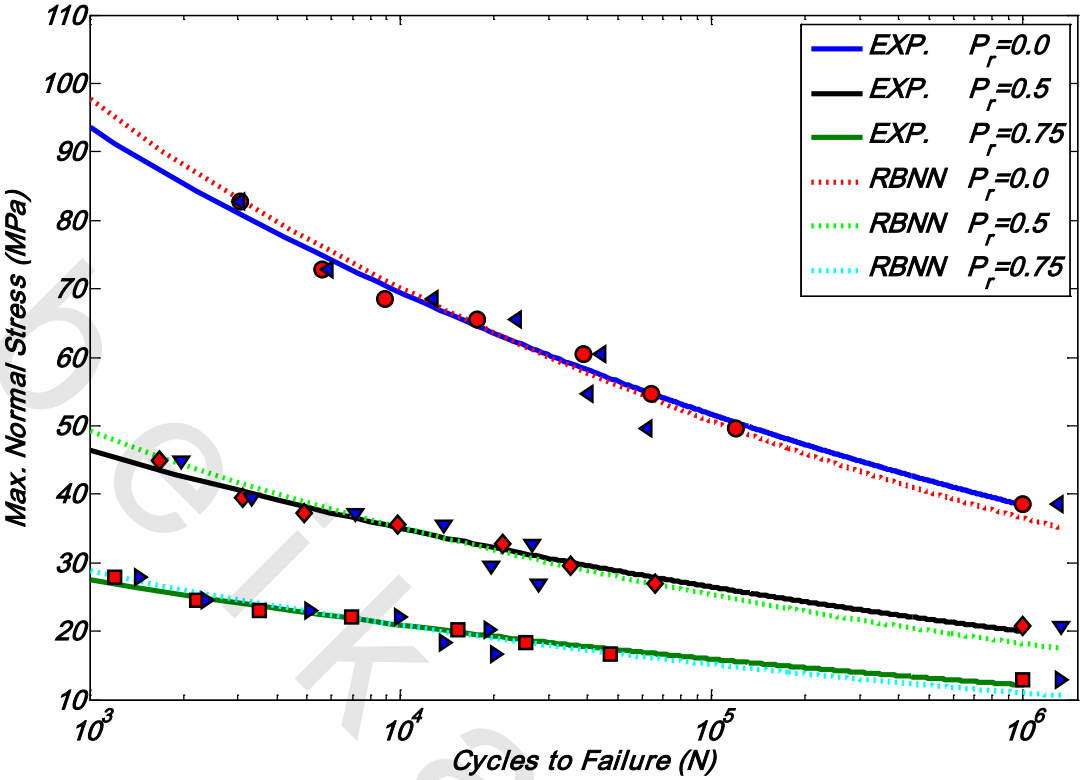


Figure 7.23 Comparison between the experimental data and the Radial basis neural network RBNN predicted data for M_1, $[\pm 45°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

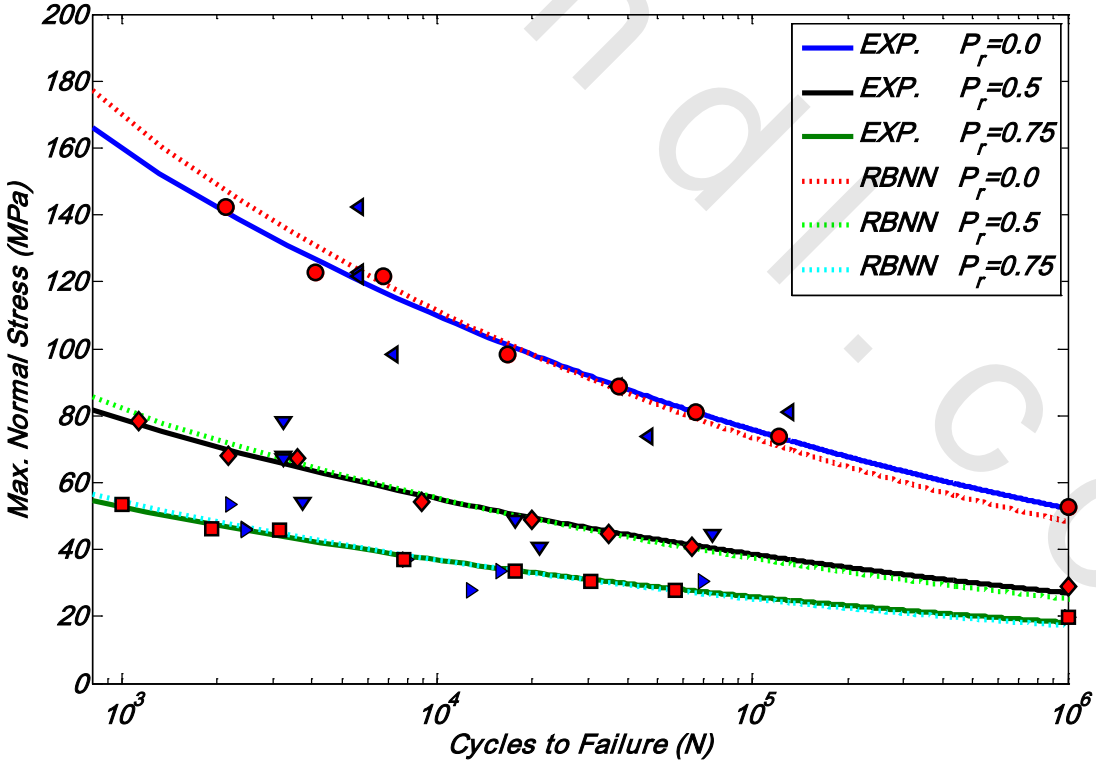


Figure 7.24 Comparison between the experimental data and the Radial basis neural network RBNN predicted data for M_2, $[0,90°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

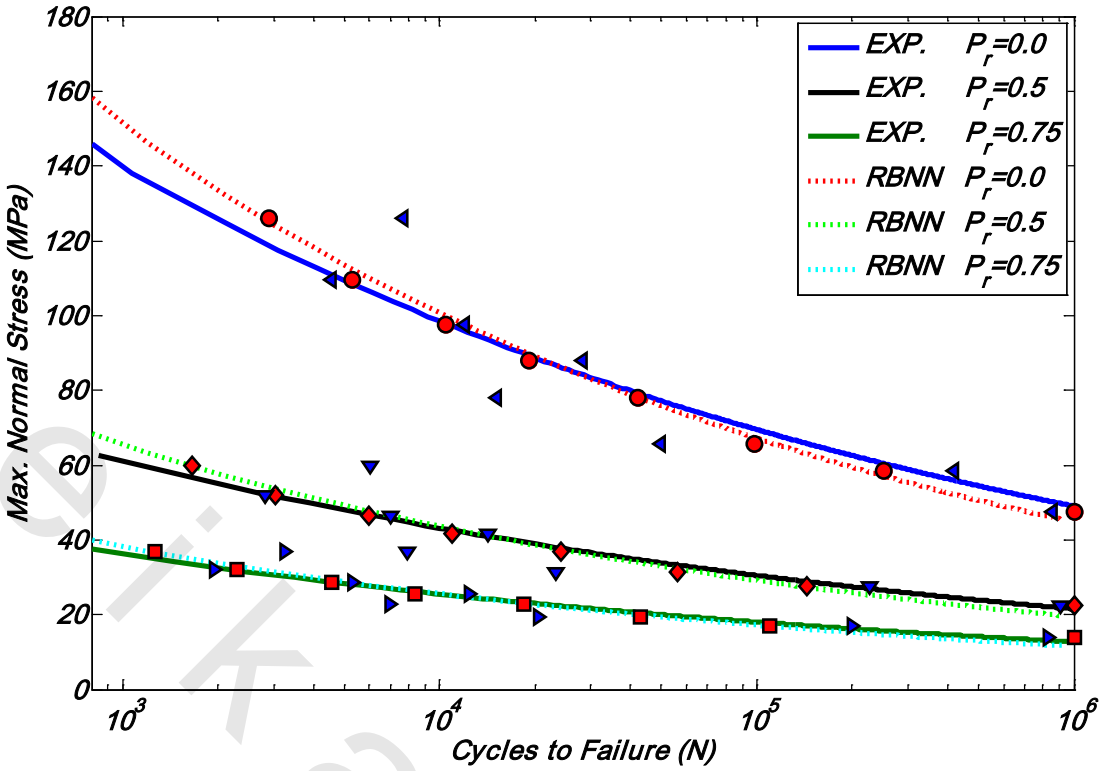


Figure 7.25 Comparison between the experimental data and the g Radial basis neural network RBNN predicted data for M_2, $[\pm 45°]_{3s}$ specimens with $P_r = 0, 0.5, 0.75$

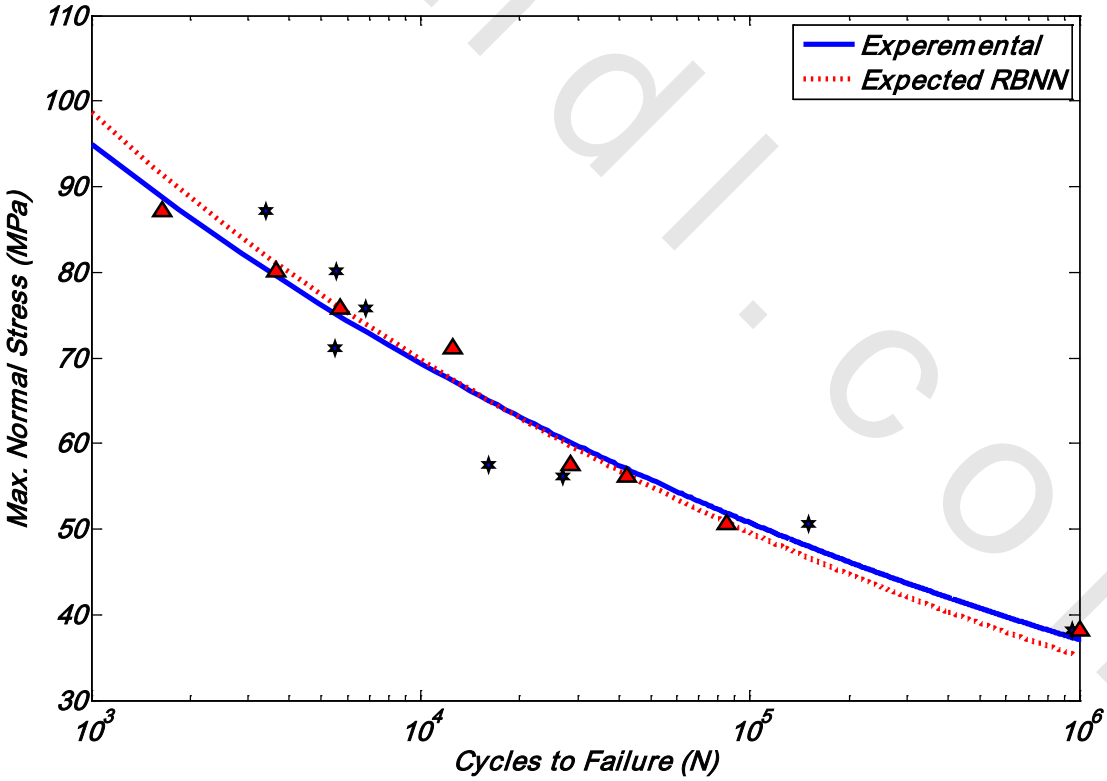


Figure 7.26 Comparison between the experimental data and the Radial basis neural network RBNN Expected data for M_1, $[0,90°]_{3s}$ specimens with $P_r = 0.25$

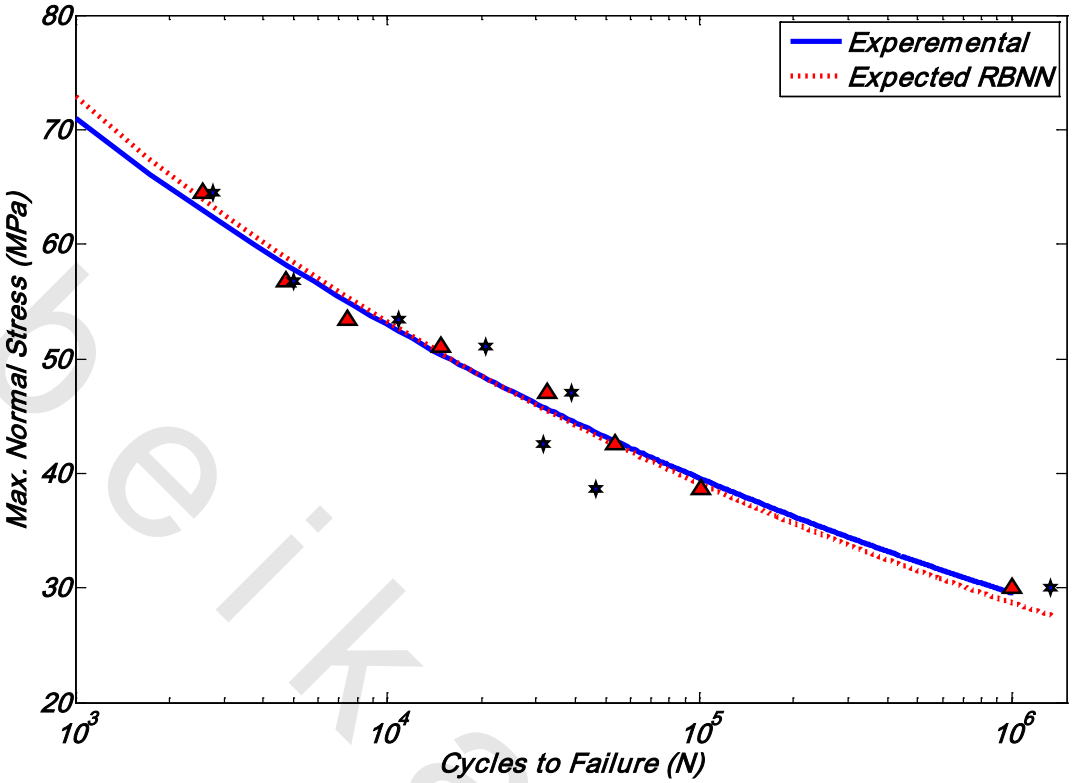


Figure 7.27 Comparison between the experimental data and the Radial basis neural network RBNN Expected data for M_1, $[\pm 45°]_{3s}$ specimens with $P_r = 0.25$

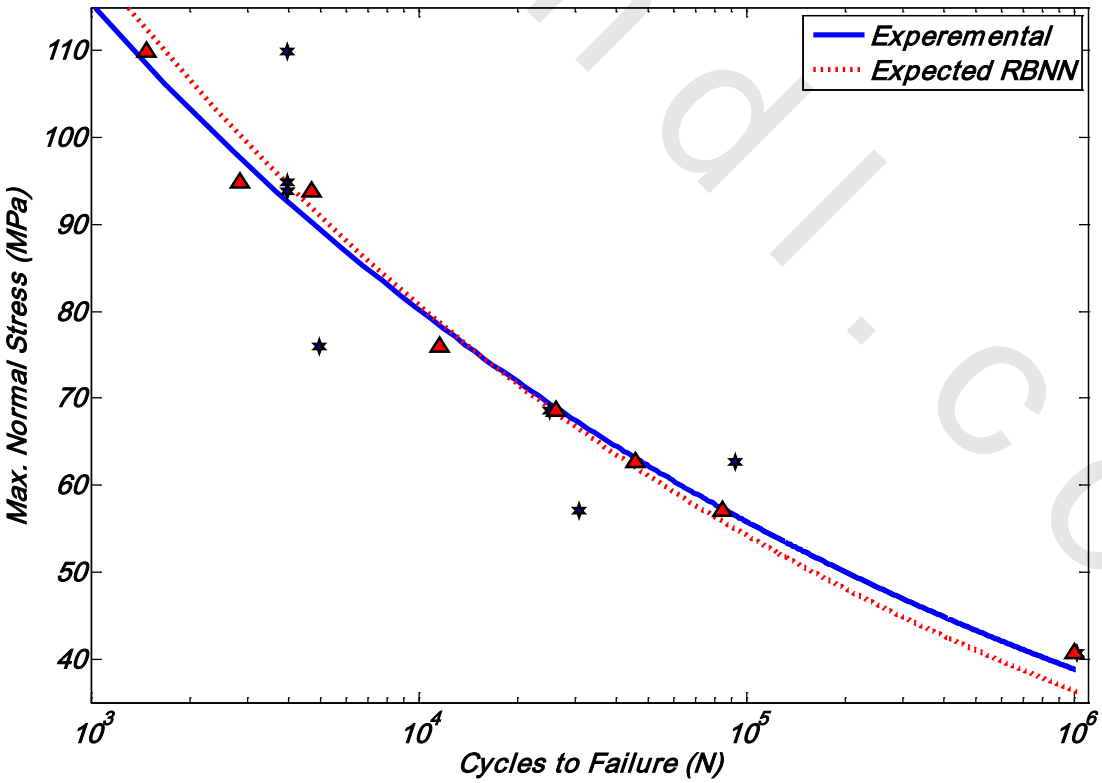


Figure 7.28 Comparison between the experimental data and the Radial basis neural network RBNN Expected data for M_2, $[0,90°]_{3s}$ specimens with $P_r = 0.25$

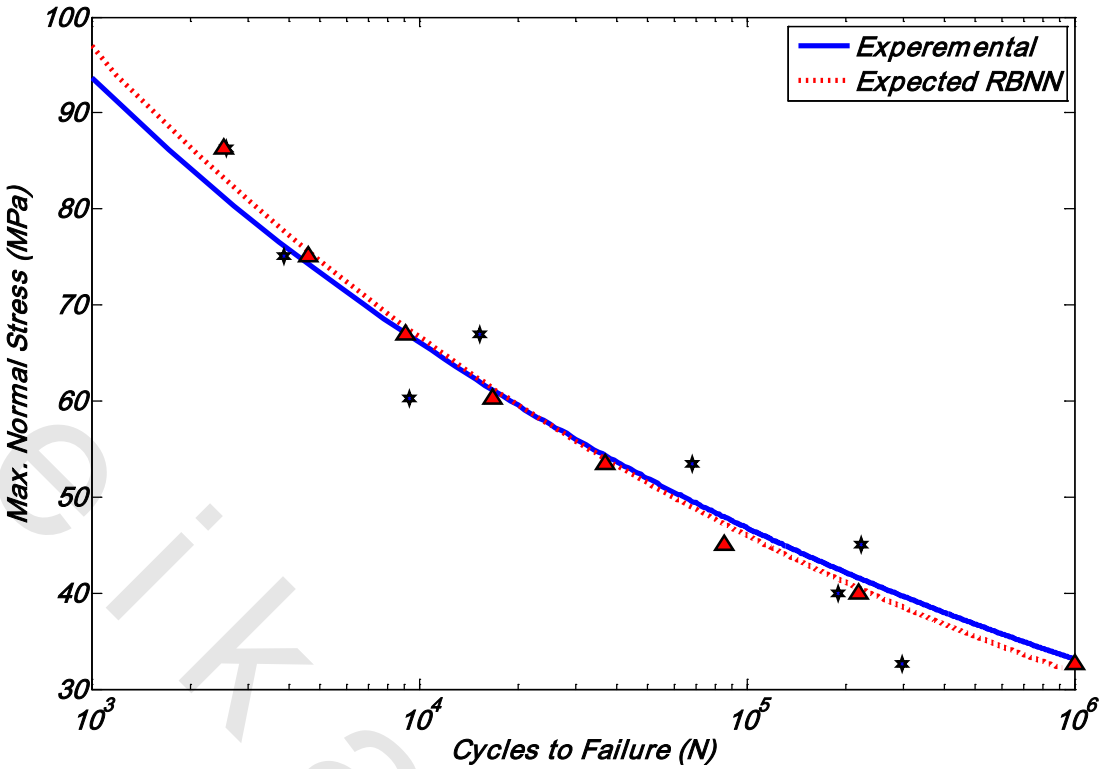


Figure 7.29 Comparison between the experimental data and the Radial basis neural network RBNN Expected data for M_2, $[\pm 45°]_{3s}$ specimens with $P_r = 0.25$

7.8 Neural System Validation and Reliability

It is very useful from the designer point of view to have an neural system aids to decide whether his suggested design for composite structure is suitable or not by compute the percentage of error for fatigue constant (a) and the Mean Square error, MSE.

Table 7.1 gives the values of percentage of error for fatigue constant (a), between the expected and experimental data for all networks with $P_r = 0.25$.

Table 7.2 gives the values of mean square error (MSE), from Equation 7.1, between the expected and experimental data for all networks in order to compare between the performances of all networks.

$$MSE = \frac{\sum(N_{nn} - N)^2}{n} \qquad (7.1)$$

Where: N_{nn} is the predicted number of cycles to failure,

N is the number of cycles to failure measured from experimental work,

n is the number of experimentally measured data values.

Table 7.1: percentage error of fatigue constant (a) values at $P_r = 0.25$

Fiber orientation	Method of Manufacturing	a [MPa]			b			Error in (a) %		
		FFNN	GRNN	RBNN	FFNN	GRNN	RBNN	FFNN	GRNN	RBNN
[0,90°]	M_1	245	257	274.7	-0.1387	-0.1397	-0.1487	1.03	5.97	13.27
	M_2	336	369.1	397	-0.1563	-0.1642	-0.1729	1.72	7.66	16.11
[±45°]	M_1	171.3	177	183	-0.1273	-0.1303	-0.134	0.94	4.3	7.83
	M_2	266	280.5	294.6	-0.1512	-0.1558	-0.1613	1.02	6.53	11.88

Table 7.1: Mean square error (MSE) values at $P_r = 0.25$

Fiber orientation	Method of Manufacturing	MSE		
		FFNN	GRNN	RBNN
$[0,90°]_{3s}$	M_1	6.8 E-3	1.41 E-2	5.37 E-2
	M_2	7.5 E-3	1.59 E-2	7.71 E-2
$[\pm45°]_{3s}$	M_1	6 E-3	1.26 E-2	2.95 E-2
	M_2	6.7 E-3	1.32 E-2	6.34 E-2

7.9 Conclusions

According to the above simulation it is concluded that:

1) The feed forward neural network, FFNN, generalized regression neural network, GRNN, and radial basis neural network, RBNN, are suitable for life prediction of GFRE.
2) The results show much better predication quality for the case of $[\pm45°]_{3s}$ than $[0,90°]_{3s}$. The mean square error between experimental data and predicted data for the case of $[\pm45°]_{3s}$ is less than $[0,90°]_{3s}$, for feed forward neural network, FFNN, generalized regression neural network GRNN, and radial basis neural network, RBNN.
3) The results show much better predication quality for the manufacture method of M_1 than M_2. The mean square error between experimental data and predicted data for the case of M_1 is less than M_2, for feed forward neural network, FFNN, generalized regression neural network, GRNN, and radial basis neural network, RBNN.
4) Feed forward neural network, FFNN, is more suitable than the generalized regression neural network, GRNN, and radial basis neural network, RBNN, for representing case study of GFRE by giving smallest mean square error.

7.10 The Use of Present Artificial Neural Network in Predicting Non-experimental Data

The main goal of the artificial neural network design is predicting non-experimental data. In this section we will use the suggested feed forward neural network, FFNN, to predict some non-experimental data not included in experimental tests, because it has minimum mean square error (MSE). It is selected to use three different values of the pressure ratios (P_r) of 0.3, 0.6 and 0.9 and random values of maximum normal stresses (σ_{max}) for both fiber orientations, $[0,90°]_{3s}$ and $[\pm45°]_{3s}$, and both methods of manufacturing M_1 and M_2. The previous four parameters are the input vectors for artificial neural network, while the output is the signal vector; the number of cycles to failure (N).

Figures 7.30 to 7.33 represent the maximum normal stresses (σ_{max}) against the number of cycles to failure (N) output from artificial neural network for both fiber orientations and both methods of manufacturing. Using the power formula $\sigma_{max} = aN^b$ have proved its suitability by giving acceptable values for the correlation factors. Comparing the result predicted from artificial neural network and data obtained from experimental tests for the other values of pressure ratios (P_r), Tables 7.3 to 7.6; it is conclude that the present artificial neural network is suitable and useful in predicting non-experimental data.

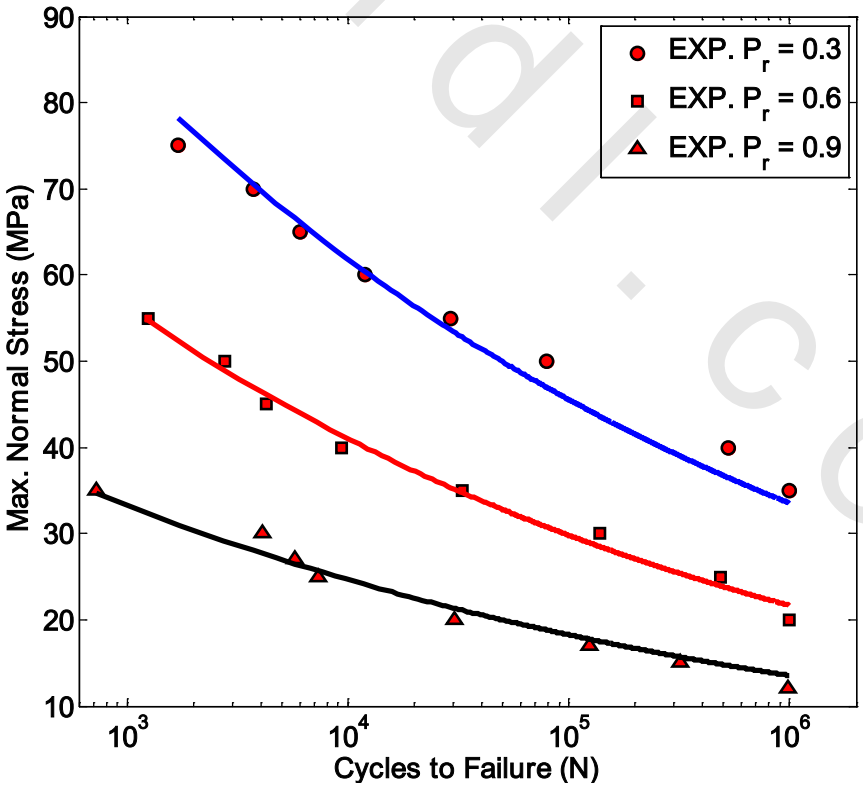


Figure 7.30 Expected Data for M_1, $[0,90°]_{3s}$ specimens

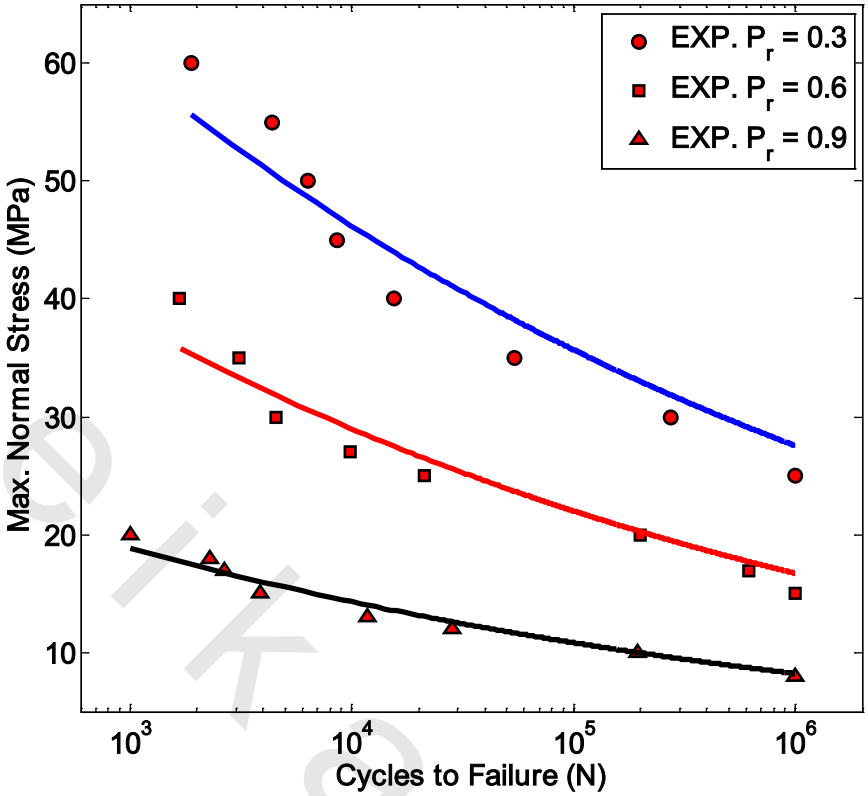


Figure 7.31 Expected Data for M_1, $[\pm 45°]_{3s}$ specimens

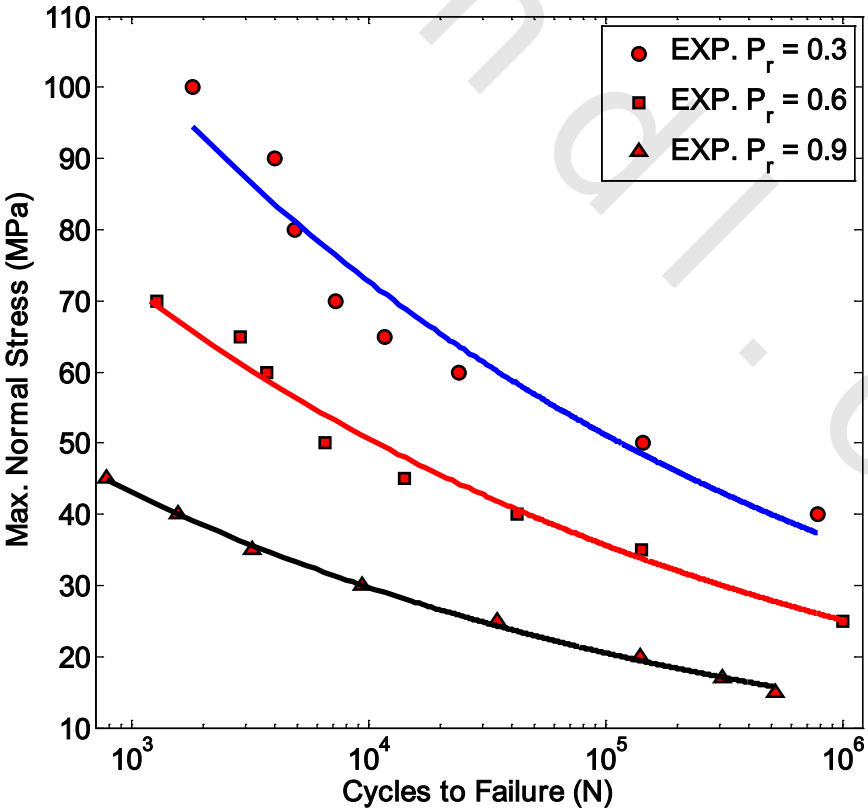


Figure 7.32 Expected Data for M_2, $[0,90°]_{3s}$ specimens

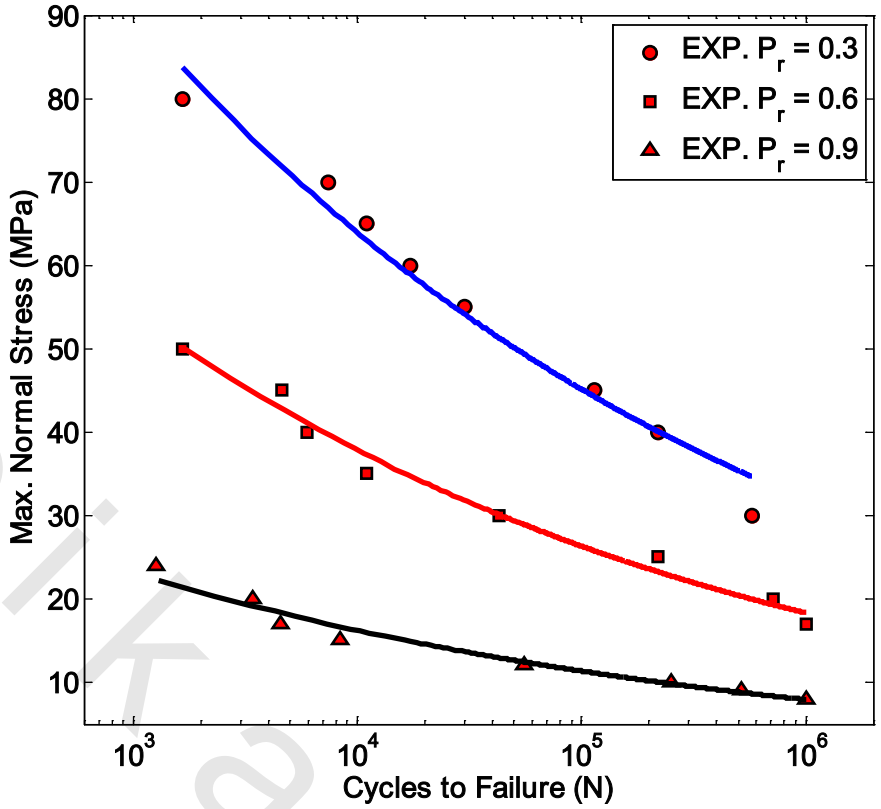


Figure 7.33 Expected Data for M_2, $[\pm 45°]_{3s}$ specimens

Table 7.3: Fatigue Constant (a) and (b) for M_1, $[0,90°]_{3s}$ specimens tested

Pressure ratio (P_r)	***[0,90°]$_{3s}$***		
	a (MPa)	***b***	***Correlation factor***
0	314.3	-0.1361	0.9926
0.25	242.5	-0.1359	0.9858
0.3*	**209.8***	**-0.1327***	**0.9728***
0.5	163.2	-0.133	0.9837
0.6*	**146.4***	**-0.1383***	**0.9901***
0.75	99.88	-0.1305	0.9811
0.9*	**81.56***	**-0.13***	**0.974***

* **Expected Data**

Table 7.4: Fatigue Constant (a) and (b) for M_1, $[\pm 45°]_{3s}$ specimens tested

Pressure ratio (P_r)	***[±45°]$_{3s}$***		
	a (MPa)	***b***	***Correlation factor***
0	226.6	-0.1284	0.9953
0.25	169.7	-0.1264	0.9876
0.3*	**129.8***	**-0.1121***	**0.9238***
0.5	107.5	-0.1218	0.9845
0.6*	**86.8***	**-0.1191***	**0.9373***
0.75	62.09	-0.1181	0.9803
0.9*	**43.22***	**-0.12***	**0.9596***

*** Expected Data**

Table 7.5: Fatigue Constant (a) and (b) for M_2, $[0,90°]_{3s}$ specimens tested

Pressure ratio (P_r)	***[0,90°]$_{3s}$***		
	a (MPa)	***b***	***Correlation factor***
0	484.4	-0.1612	0.9924
0.25	341.9	-0.1574	0.9911
0.3*	**296.3***	**-0.1526***	**0.9386***
0.5	229.2	-0.1548	0.9891
0.6*	**205.8***	**-0.1523***	**0.9743***
0.75	151.6	-0.1535	0.9878
0.9*	**130.7***	**0.1609***	**0.9978***

*** Expected Data**

Table 7.6: Fatigue Constant (a) and (b) for M_2, $[\pm 45°]_{3s}$ specimens tested

Pressure ratio (P_r)	***[±45°]$_{3s}$***		
	a (MPa)	***b***	***Correlation factor***
0	397.2	-0.1514	0.9843
0.25	263.3	-0.1501	0.9832
0.3*	**256.2***	**-0.1508***	**0.9727***
0.5	173.6	-0.1511	0.9837
0.6*	**160.1***	**-0.1568***	**0.9833***
0.75	102.7	-0.1512	0.9816
0.9*	**67.78***	**-0.1553***	**0.9636***

*** Expected Data**

In order to examine the suitability of the present artificial neural network and in the same time to check the validity of the present modified Goodman's equation and new failure criterion presented in sections (6.3) and (6.6.3) respectively, we will use the data predicted from the present artificial neural network for substituting in these equations.

Figure 7.34 show that the Goodman's R.D. gives excellent results for both manufacturing method M_1 and M_2 for all fiber orientations and for all expected pressure ratios data from present artificial neural network, where, the values of Goodman's R.D. are around the theoretical value of unity. For both manufacturing method M_1 and M_2 for all fiber orientations, the Goodman's R.D. are ranging from 0.9042 as a minimum value to 1.2255 as a maximum value with an average value of 1.0547 and standard deviation of 0.0793 which are very near to unity, and the difference may be referred to scatter in experimental data.

Figure 7.35 show that the modifying criterion gives excellent results for the data expected. The values of R.D. for both manufacturing method M_1 and M_2 for all fiber orientations, the R.D. are ranging from 0.86696 as a minimum value to 1.1839 as a maximum value with an average value of 1.0718 and standard deviation of 0.07515.

Finally, from the data predicted and expected by NN designed, we can be concluded; the NN system provided the designer with reliable results to help him to decide whether the choice of the combination of σ_{max}, P_r, fiber orientation and number of cycles to failure N, is correct or not.

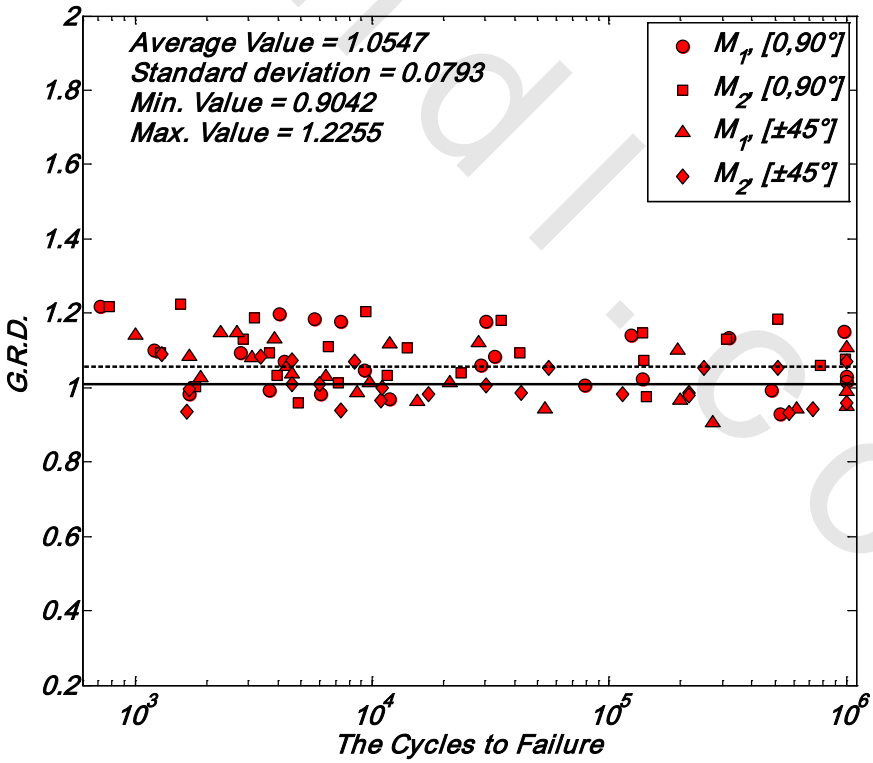


Figure 7.34 The G.R.D. of both manufacturing method M_1 and M_2 for all fiber orientations and all pressure ratios for expected data

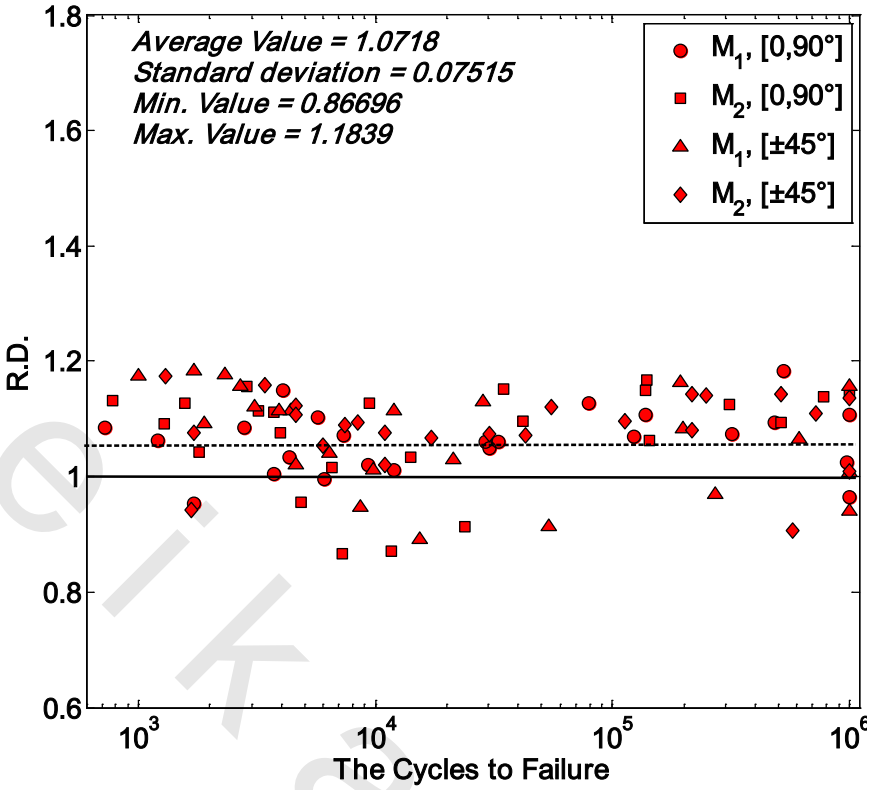


Figure 6.35 Relative damage (R.D.) applying present failure criterion of both manufacturing method M_1 and M_2 for all fiber orientations and all pressure ratios for expected data