Induction Machine Fault Detection Using Support Vector Machine Based Classifier

V.N.Ghate Electrical Engineering Department Government College of Engineering, Amravati (MS) INDIA <u>vng786@rediffmail.com</u> http://www.gcoea.ac.in

S.V.Dudul Applied Electronics Department Sant Gadge Baba Amravati University,Amravati INDIA dudulsv@rediffmail.com

Abstract: Industrial motors are subject to various faults which, if unnoticed, can lead to motor failure. The necessity of incipient fault detection can be justified by safety and economical reasons. The technology of artificial neural networks has been successfully used to solve the motor fault detection problem. This paper develops inexpensive, reliable, and noninvasive NN based fault detection scheme for small and medium sized induction motors. Detailed design procedure for achieving the optimal NN model and Principal Component Analysis for dimensionality reduction is proposed. Overall thirteen statistical parameters are used as feature space to achieve the desired classification. Generalized Feed Forward (GFFDNN) and Support Vector Machine (SVM) NN models are designed and verified for optimal performance in fault identification on experimental data set of custom designed 2 HP, three phase 50 Hz induction motor.

Keywords: Induction motor, Fault detection, Neural Network, GFFDNN, SVM, PCA

1 Introduction

Induction machines play a crucial role in certain industries, such as manufacture, transportation, etc. They offer the core capabilities for industrial success and the maintenance of them is essential and profitable to most electrical industrial processes. A lack of coherent maintenance strategy may lead to the loss of individual items of a plant, and a heavy capitalized losses burden. As it is not economical to introduce redundant backup machines, online monitoring for induction machines is important for safe operation and production quality. In order to keep machines in good condition, techniques such as fault monitoring, detection, classification, and diagnosis have become increasingly essential [1]-[3]. There are invasive and noninvasive methods for machine fault detection [4], [5], [7] The noninvasive methods are more preferable than the invasive methods because they are based on easily accessible and inexpensive measurements to diagnose the machine conditions without disintegrating the machine structure. Recently, artificial intelligence (AI) techniques have been proposed for the noninvasive machine fault detection [4], [6], [8].

They have several advantages over the traditional model-based techniques [6], [9]. They require no detailed analysis of the different kinds of faults or modeling of the system. These AI-based techniques include expert systems, neural network, and fuzzy logic. An expert system is able to manage knowledge-based production rules that model the physical system [11], [12]. Neural network approaches can be considered as "black-box" methods as they do not provide heuristic reasoning about the fault detection process [4], [5], [10]. Fuzzy logic systems can heuristically implement fault detection principles and heuristically interpret and analyze their results [13], [14], [15]. In this paper, neural network type approach is used because feed forward neural generalized network (GFFDNN) is able to provide an accurate fault diagnostic classification.

2 Feature Extraction

The main problems facing the use of ANN are the selection of the best inputs and how to choose the ANN parameters making the structure compact, and creating highly accurate networks. For the proposed system, the feature selection is also an important process since there are many features after feature extraction. Many input features need significant computational efforts to calculate, and may result in a low success rate. In order to collect data at different conditions i.e. healthy condition, under inter turn fault, Eccentricity and both i.e .inter turn and Eccentricity specially designed 2 HP, 4 pole, 415V, 50 Hz ,three phase induction motor is used. Three AC current probes were used to measure the stator current signals. From the time waveforms, as shown in Fig.1. no conspicuous difference exists among the different conditions.



Fig.1.Experimental waveform for current of phase A for various conditions

There is a need to come up with a feature extraction method to classify faults. To classify the different faults the statistical parameters are used. To be precise, 'sample' statistics will be calculated for current data. Overall thirteen parameters are calculated as input feature space. Minimum set of statistic to be examined includes the root mean square (RMS) of the zero mean signal (which is the standard deviation), the maximum, and minimum values the skewness coefficient and kurtosis coefficient. *Pearson's coefficient of skewness*, g_2 defined by:

$$g_2 = \frac{3\left(\bar{x} - \tilde{x}\right)}{S_x} \tag{1}$$

Where x denotes mean, x denotes median and S_x denotes the sample standard deviation. The sample coefficient of variation v_x is defined by;

$$v_x = \frac{S_x}{\overline{x}} \tag{2}$$

The r^{th} sample moment about the sample mean for a data set is given by;

$$m_{r} = \frac{\sum_{i=1}^{n} \left(x_{i} - \overline{x} \right)^{'}}{n}$$
(3)

 m_2 denotes to spread about the center, m_3 refers to skewness about the center; m_4 denotes to how much data is massed at the center. Second, third and

fourth moments are used to define the sample coefficient of skewness, g_3 and the sample coefficient of kurtosis, g_4 as follows.

$$g_{3} = \frac{m_{3}}{\left(\sqrt{m_{2}}\right)^{3}}$$
(4)

$$g_{4} = \frac{m_{4}}{\left(\sqrt{m_{2}}\right)^{4}} \tag{5}$$

The sample covariance between dimensions j and k is defined as;

$$c_{jk} = \frac{\sum_{i=1}^{n} (x_{ij} - \overline{x_{j}}) (x_{ik} - \overline{x_{k}})}{(n-1)}$$
(6)

The ordinary correlation coefficient for dimensions j and k, r_{jk} is defined as;

$$r_{jk} = \frac{C_{jk}}{S_j - S_k} \tag{7}$$

3 Generalized Feed Forward NN

Generalized feed forward network is a generalization of the MLP such that connections can jump over one or more layers. In this network it is allowed to cross the hidden layer, i.e. output layer will get the input from the hidden layer and directly from input layer also.

Generalized feed forward Neural Network is proposed as fault classifier. Number of input Processing Elements (PE) must be equal to that of number of input statistical parameters so 13 input Processing Elements are used in input layer. Four Processing Elements are used in output layer for four conditions of motor namely Healthy, Inter turn fault, Eccentricity and Both faults. For data processing MATLAB7.1, Neuro Solution 5.0 and XLSTAT is used. General learning algorithm used is as follows:

Initialization of Weights:

Step 1: Initialize the weights to small random values Step 2: While stopping condition is false, do step 3-10

Step 3: For each training pair do steps 4-9

Feed forward:

Step 4: Each input unit receives the input signal x_i and transmits this signals to all units in the hidden layer

Step 5: Each hidden unit $(z_j, j=1,...,p)$ sums its weighted input signals

$$z_{-inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$
(8)

Applying the activation function $Z_j = f(z_{inj})$ here the activation function is

 $tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ and sends this signal to all units in output units.

Step 6: Each output unit $(y_k, k=1,...,m)$ sums its weighted input signals,

$$y_{-ink} = w_{ok} + \sum_{j=1}^{p} z_i w_{jk} + \sum_{j=1}^{p} x_i v_{ij}$$
(9)

And applies its activation function to calculate the output signals $Y_k = f(y_{-ink})$ here the activation function is

$$\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x}) \quad (10)$$

Back Propagation Error:

Step 7: Each output unit $(y_k, k=1,...,m)$ receives a target pattern corresponding to an input pattern error information term is calculated as

$$\boldsymbol{d}_{k} = \left(t_{k} - y_{k}\right) f\left(y_{-ink}\right) \tag{11}$$

Step 8: Each hidden unit $(z_j, j=1,...,p)$ sums its delta inputs from units in the layer above

$$\boldsymbol{d}_{-inj} = \sum_{k=1}^{m} \boldsymbol{d}_{i} \boldsymbol{w}_{jk}$$
(12)

The error information term is calculated as

$$\boldsymbol{d}_{j} = \boldsymbol{d}_{-inj} f\left(\boldsymbol{z}_{-inj}\right) \tag{13}$$

Updation of weight and Biases:

Step 9: Each output unit $(y_k, k=1,...,m)$ updates its bias and weights (j=0,...,p)

$$w_{jk}(t+1) = w_{jk}(t) + ad_k z_j + m[w_{jk}(t) - w_{jk}(t-1)]$$
(14)

Where \boldsymbol{a} is learning rate and \boldsymbol{m} is momentum factor

And each hidden unit (z_j , j=1,...,p) updates its bias and weights (i=0,...,n)

$$v_{jk}(t+1) = v(t) + \mathbf{ad}_{j}x_{i} + \mathbf{m}[v_{ij}(t) - v_{ij}(t-1)]$$
(15)
Step 10: Test the stopping condition

Selection of Error criterion:

Supervised learning requires a metric, a measure of how the network is doing. Members of the Error Criteria family monitor the output of a network, compare it with some desired response and report any error to the appropriate learning procedure. In gradient descent learning, the metric is determined by calculating the sensitivity that a cost function has with respect to the network's output. This cost function, J, is normally positive, but should decay towards zero as the network approaches the desired response. The literature has presented several cost functions, in which p is to be define such as p=1, 2, 3, 4... 8 criterion is $L_1, L_2, L_3, L4 \dots L_8$

Components in the ErrorCriteria family are defined by a cost function of the form:

$$J(t) = \frac{1}{2} \sum_{i=1}^{p} \left(d_i(t) - y_i(t) \right)^p$$
(16)

and error function:

$$e_i(t) = (d_i(t) - y_i(t))$$
 (17)

Where d(t) and y(t) are the desired response and network's output, respectively. To select the correct error criterion various error criterion has been tested and results are shown in Fig. 2, Fig.3 and Fig. 4.



Criterion



For selection of hidden layer PEs randomize data is fed to the neural network and is retrained five times with different random weight initialization. It is observed that network with single hidden layer and 5 PEs in hidden layer gives the better results as shown in Fig.5(a) and Fig.5(b).



Y: Number of PEs in Hidden Layer X: No. of Epochs Fig.5 (a) Variation of Average MSE with Number of PEs in Hidden Layer



Fig.5 (b) Variation of Average Minimum MSE with Number of PEs in Hidden Layer

Various transfer functions and learning rules namely Momentum (MOM), Conjugate-Gradient (CG), Quick Propagation (QP), Delta Bar Delta (DBD), Levenberg-Marquardt (LM) and Step (STP) are verified for training and testing the network. Average minimum MSE on training and CV data and classification accuracy on testing, CV and training data is compared in Table 1 and Table 2 (Appendix)

The parameters of the hidden layer and output layer i.e. step size and momentum are selected by comparing average minimum MSE. In Hidden layer optimum value of Step size is 0.13 and momentum is 0.6 and for output layer Step size is 0.05 and momentum is 0.08. Performance is shown in Fig.6, Fig. 7 and

Fig.8.



Fig.6 Variation of Average Minimum MSE with Step size in Hidden Layer



Fig.7 Variation of Average Minimum MSE with Momentum rate in Hidden Layer



Fig.8 Variation of Average Minimum MSE with Step size and Momentum rate in output Layer

From above experimentation, selected parameters for GFFD-NN are given below.

GFFD- NN (13-5-4), Number of epochs = 5000,

Exemplars for training = 70%,

Exemplars for cross validation = 15%,

Exemplars for Testing = 15%

Number of Hidden Layers: 01

T.F.: Tanh	Learning Rule: Momentum
Step size: 0.13	Momentum: 0.6
Output Layer:	
T.F.: Tanh	Learning Rule: Momentum
Step size: 0.05	Momentum: 0.08
Number of connection v	weights: 146

Time Elapsed per epoch per exemplar: 1.001 msec.

Different datasets are formed using variable split ratios and leave-N-out cross validation technique. Proposed NN is trained and tested five times on various datasets and later validated carefully so as to ensure that its performance does not depend on specific data partitioning scheme. The performance of the NN should be consistently optimal over all the datasets with respect to MSE and classification accuracy. To check the learning ability and classification accuracy the total data is divided in four groups. First two groups (50% data) are tagged as Training data and third and forth group (each 25%) is tagged for Cross Validation and Testing (1234:1,2-TR, 3-CV, 4-Test). Similar 18 combinations are prepared and network is train and test for each group. Results are shown in Fig.9 to Fig. 12.



Fig. 9 Variation of average Minimum MSE with Test on Testing and Training dataset and percent data tagged for training







Fig. 11 Variation of average MSE with Test on Testing, CV and Training dataset with CV rows shifted (n)



Fig. 12 Variation of average Minimum MSE with Training and CV with group of Dataset

4 Support Vector Machine

The support vector machine (SVM) in is a new kind of classifier that is motivated by two concepts. First, transforming data into a high-dimensional space can transform complex problems (with complex decision surfaces) into simpler problems that can use linear discriminant functions. Second, SVMs are motivated by the concept of training and using only those inputs that are near the decision surface since they provide the most information about the classification.

It is a kind of learning machine based on statistical learning theory. The basic idea of applying SVM to pattern classification can be stated as follows: first map the input vectors into one features space, possible in higher space, either linearly or nonlinearly, which is relevant with the kernel function. Then, within the feature space from the first step, seek an optimized linear division, that is, construct a hyperplane which separates two classes. It can be extended to multi-class. SVMs training always seek a global optimized solution and avoid over fitting, so it has ability to deal with a large number of feature.

Kernel Adatron algorithm:

For N dimensional space data x_i (i = 1...N) this algorithm can be easily extended to network by substituting the inner product of patterns in the input space by the kernel function, leading to the following quadratic optimization problem:

$$J(\mathbf{a}) = \sum_{i=1}^{N} \mathbf{a}_{i} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{a}_{i} \mathbf{a}_{j} d_{i} d_{j} G(x_{i} - x_{j}, 2\mathbf{s}^{2})$$
(16)

Subject to

$$\sum_{i=1}^{N} d_i \boldsymbol{a}_i = 0 \qquad \boldsymbol{a}_i \ge 0, \forall i \in \{1, ..., N\}$$
(17)

where $G(x, s^2)$ represents a Gaussian function, N is the number of samples, a_i are a set of multipliers (one for each sample),

$$J(x_i) = d_i (\sum_{i=1}^N d_j a_j G(x_i - x_j, 2s^2) + b)$$
 (18)

and

$$M = \min g(x_i) \tag{19}$$

and choose a common starting multiplier \mathbf{a}_i , learning rate ?, and a small threshold. Then, while M > t, we choose a pattern x_i and calculate an update $\Delta \mathbf{a}_i = \mathbf{h}(1 - g(x_i))$ and perform the update If $\mathbf{a}_i(n) + \Delta \mathbf{a}_i > 0$

$$\mathbf{a}_{i}(n+1) = \mathbf{a}_{i}(n) + \Delta \mathbf{a}_{i}(n)$$

$$b(n+1) = b(n) + d_{i}\Delta \mathbf{a}_{i}$$
(20)

And if $\boldsymbol{a}_i(n) + \Delta \boldsymbol{a}_i \leq 0$

$$\boldsymbol{a}_{i}(n+1) = \boldsymbol{a}_{i}(n)$$

$$\boldsymbol{b}(n+1) = \boldsymbol{b}(n) \tag{21}$$

After adaptation only some of the a_i are different from zero (called the support vectors). They correspond to the samples that are closest to the boundary between classes. This algorithm can be considered the "on-line" version of the quadratic optimization approach utilized for SVMs, and it can find the same solutions as Vapnik's original algorithm for SVMs. It is easy to implement the kernel Adatron algorithm since $g(x_i)$ can be computed bcally to each multiplier, provided that the desired response is available in the input file. In fact, the expression for $g(x_i)$ resembles the multiplication of an error with an activation, so it can be included in the framework of neural network learning. The Adatron algorithm essentially prunes the RBF network so that its output for testing is given by,

$$f(x) = \operatorname{sgn}(\sum_{\substack{i \in spport\\vectors}}^{N} d_i \boldsymbol{a}_i G(x_i - x_i, 2\boldsymbol{s}^2) - b) \quad (22)$$

And cost function in error criterion is

$$J(t) = \frac{1}{2} \sum_{i=1}^{2} \left(d_i(t) - (\tanh(y_i(t)))^2 \right)$$
(23)

For selection of step size randomize data is fed to the neural network and is retrained five times with different random weight initialization. It is observed that 0.8 step size gives the optimal result. Number of connection weights: 472

Time Elapsed per epoch per exemplar: 0.934 ms



Fig.13 Variation of Average Minimum MSE with Step size



Fig. 14 Variation of Average MSE with Step size

Using the similar datasets SVM classifier is tested retaining five times and results are shown in Fig.15 to Fig.18



Fig. 15 Variation of average Minimum MSE with Test on Testing and Training dataset and percent data tagged for training



Fig. 16 Variation of average Classification Accuracy with testing on Testing and Training dataset and percent data tagged for

training



Fig. 17 Variation of average MSE with Test on Testing, CV and Training dataset with CV rows shifted (n)



Fig. 18 Variation of average Minimum MSE with Training and CV with group of Dataset

5 Dimensionality Reduction Using Principal Component Analysis

One problem appears after the feature extraction. There are too many input features that would require a significant computational efforts to calculate, and may result in low accuracy of the monitoring and fault diagnosis. The potential improvements which can be achieved by first mapping the data into a space of lower dimensionality. Reduction in dimensionality of the input space and hence the network can be achieved by Principal Component Analysis (PCA). PCA is performed by Pearson rule. The Fig.12 is related to a mathematical object, the eigenvalues, which reflect the quality of the projection from the 13-dimensional to a lower number of dimensions.



variability.

Using the results of Principal Component Analysis, dimensions of GFFD-NN can be reduced. Number of inputs are reduced to five. By similar experimentations, the optimum GFFD-NN classifier is designed with the following changes;

Number of Inputs = 5, Number of PEs in Hidden Layer = 5, Number of epochs = 5000, Exemplars for

training = 70%,

Exemplars for cross validation = 15%,

Exemplars for Testing = 15%,

Number of Hidden Layers: 01

T.F.: Tanh Learning Rule: Momentum Step size: 0.09 Momentum: 0.6

Output Layer:

T.F.: Tanh Learning Rule: Momentum Step size: 0.05 Momentum: 0.08

Number of connection weights: 74

Time Elapsed per epoch per exemplar: 0.786 ms Training and testing results for new model is as shown in Fig.20 to Fig.25



Number of Rows Shifted for Cross Validation (n) Fig. 24 Variation of average MSE with Test on Testing, CV and Training dataset with CV rows shifted (n)



Fig. 25 Variation of average Minimum MSE with Training and CV with group of Dataset

Using the results of PCA, dimensions of the support vector machine classifier are also reduced. It is found that number of inputs, reduced to five and step size of 0.7 gives the optimal results. Number of connection weights: 264

Time Elapsed per epoch per exemplar: 0.693 ms

Variation of average minimum MSE and average classification accuracy with number of PCs as input is shown in Fig. 26 and Fig.27. and training and testing results are shown in Fig.28 and Fig.29.



Y: No. of PCs X: No. of Epochs Fig. 26 Variation of Average MSE of training and CV on number of PCs as inputs



Fig. 27 Variation of average Minimum MSE and average classification accuracy with number of PCAs as input



Fig. 28 Variation of average MSE with Test on Testing, CV and Training dataset with CV rows shifted (n)



Fig. 29 Variation of average Minimum MSE with Training and CV with group of Dataset

6 Robustness of Classifier to Noise

Since the proposed classifier is to be used in real time. where measurement noise is anticipated, it is necessary to check the robustness of classifier to noise. To check the robustness Uniform and Gaussian noise with mean value zero and variance varies from 1 to 20 % is introduced in input and output and average classification accuracy on testing data i.e. unseen data is checked. It is observed that in GFFDNN, average classification accuracy is not affected by both noise in input and output in SVM classification accuracy and is consistent with noise of variance up to 15%. Comparative results are shown in Table 3.

7 Results and Discussion

In this paper, the authors evaluated the performance of the developed GFFD NN and Support Vector Machine (SVM) based classifier for detection of four conditions of three phase induction motor and examined the results. After completion of the training, the learned network is able to detect different types of faults. For GFFDNN various learning rules and transfer functions are investigated for different number of hidden layers and processing elements in hidden layer. It is observed that Momentum learning rule and Tanh transfer function gives the optimal results in hidden and output layer. By varying the step size optimum results are obtained in SVM classifier. By performing Principal Component Analysis, number of inputs are reduced from 13 to 5 and thus significant reduction in dimension is achieved. From the analysis, it is seen that dimensionally reduced support vector machine(SVM-DR) based classifier works as an elegant classifier for fault diagnosis of three phase induction motor, in the sense that, average MSE on testing and cross validation samples is consistently observed as reasonably low such as 0.0591 and 0.0619, respectively. In addition, average classification accuracy on testing as well as cross validation instances is obtained as 99.61% and 98.72%, respectively indicating a reasonable classification. Also proposed classifier is enough robust to the noise, in the sense that classifier gives consistent results for Uniform and Gaussian noise with 12% variance in input and with 20% variance in output. Comparative results are shown in Table 4.

References:

- R. Isermann, "Supervision, fault-detection and fault-diagnosis methods-An introduction," *Control Eng. Practice*, vol. 5, no. 5, pp.639– 652, 1997.
- [2] [2] S. Leohardt and M. Ayoubi, "Methods of fault diagnosis," *Control Eng.Practice*, vol. 5, no. 5, pp. 683–692, 1997.
- [3] R. Patton, P. Frank, and R. Clark, *Fault Diagnosis in Dynamic Systems, Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [4] M.-Y Chow, Methedologies of using Neural Network and Fuzzy Logic for Motor Incipient Fault Detection, Singpore: World Scientific, 1997.
- [5] M. -Y. Chow, R. N. Sharpe, and J. C. Hung, "On the application and design consideration of artificial neural network fault detectors," *IEEETrans. Ind. Electron.*, vol. 40, pp. 181– 198, Apr. 1993.
- [6] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, "Recent developments of induction motor drives fault diagnosis using AI techniques," *IEEE Trans. Ind. Electron.*, vol. 47, pp. 994–1004, Oct. 2000.
- [7] G. K. Singh and S. A. S. Al Kazzaz, "Induction machine drive condition monitoring and diagnostic research - a survey," *Electric Power Systems Research*, vol. 64, no. 2, pp. 145–158, 2003.

V. N. Ghate, S. V. Dudul

- [8] Jarmo Ilonen, Joni-Kristian Kamarainen, "Diagnosis Tool for Motor Condition Monitoring" IEEE Transactions Industry Applications, vol. 41, no. 4,pp 963-971 JULY/AUGUST 2005
- [9] Onel I Y , EI Hachemi Benbouzid M., "Induction Motor Bearing Failure Detection and Diagnosis: Park and Concordia Transform Approaches Comparative Study" *IEEE Trans* on Mechatronics ,vol. 13,pp 257-262 April 2008
- [10] Tian Han,1 Bo-Suk Yang, "Fault Diagnosis System of Induction Motors Based on Neural Network and Genetic Algorithm Using Stator Current Signals" Hindawi Publishing Corporation International Journal of Rotating Machinery Volume 2006, pp 1-13 -2006
- [11] M. S. Ballal , Z. J. Khan , H. M. Suryawanshi and R. L. Sonolikar "Adaptive neural fuzzy inference system for the detection of inter-turn insulation and bearing wear faults in induction motor," *IEEE Trans. Ind. Electron.*, vol. 54, pp. 250, Feb. 2007
- [12] M. Knight and S. P. Bertani "Mechanical fault detection in a medium-sized induction motor using stator current monitoring," *IEEE Trans. Energy Convers.*, vol. 20, pp. 753, Dec. 2005
- [13] Suttichai Premrudeepreechacharn', "Induction Motor Fault Detection and Diagnosis Using Supervised and Unsupervised Neural Networks" IEEE conference ICIT'OZ-2003, Bangkok, Thailand-2003
- [14] B.S. Yang, T. Han, J.L. An "ART–KOHONEN neural network for fault diagnosis of rotating machinery" Mechanical Systems and Signal Processing vol.18 ,pp 645–657,2004
- [15] Czeslaw T Kowalski and Teresa Orlowska-Kowalska," Neural networks application for induction motor faults diagnosis" Mathematics and Computers in Simulation Volume 63, issue 3-5 pp: 435 - 448, November 2003

APPENDIX

Table 1

VARIATION OF AVERAGE MINIMUM MSE AND AVERAGE CLASSIFICATION ACCURACY WITH TRANSFER FUNCTIONS

			Average									
			Classification									
TF												
		Trai	ning			CV					тр	
	Max.	Min.	Avg.	SD	Max.	Min.	Avg.	SD	1050		IK	
Tanh	0.0024	0.0016	0.0021	0.0003	0.1997	0.1148	0.1708	0.0345	100	100	100	
L-Tanh	0.0645	0.0035	0.0158	0.0273	0.2742	0.0304	0.1506	0.0877	75	100	75	
L-sig	0.3731	0.3548	0.3668	0.0084	0.3734	0.352	0.3626	0.0084	75	66.7	65.6	
Sig	0.3569	0.3557	0.3564	0.0005	0.3659	0.3609	0.3645	0.0021	50	66.7	52.2	
Lin	0.405	0.3897	0.3979	0.0059	0.4103	0.3946	0.403	0.0057	50	0	45.5	
BAX	0.2985	0.2985	0.2985	0	0.3035	0.3035	0.3035	0	21.88	50	21.9	
AX	0.1248	0.1222	0.1233	0.001	0.4304	0.2804	0.3524	0.0649	50	83.3	46.4	
Soft	0.0184	0.0144	0.0158	0.0017	0.0484	0.0429	0.0449	0.0021	100	83.3	86.2	

Tanh : hyperbolic tan L-Tanh : Linear hyperbolic tan L-sig : Linear Sigmoid Lin : Linear

Sig: Sigmoid

BAX : Biased Axon

AX : Axon

Soft: Softmax.

Table 2VARIATION OF AVERAGE MINIMUM MSE AND AVERAGE CLASSIFICATIONACCURACY WITH TRANSFER FUNCTIONS

TF		Average Classification Accuracy									
			CV					TD			
	Max.	Min.	Avg.	SD	Max.	Min.	Avg.	SD	Test	CV	IK
Moment	0.0024	0.0016	0.0021	0.0003	0.1997	0.1148	0.1708	0.0345	100	100	100
DBD	0.016	0.0109	0.0132	0.0019	0.0128	0.0028	0.0057	0.0041	50	100	64.3
QP	0.0026	0.0025	0.0026	8E-05	0.2124	0.1536	0.1828	0.0216	100	83.3	100
CG	0.0294	0.0025	0.0086	0.0116	0.3015	0.1246	0.1666	0.0756	100	83.3	100
Step	0.0037	0.0026	0.0029	0.0004	0.1657	0.1356	0.15	0.011	100	83.3	100
LM	0.0085	0.006	0.0075	0.001	0.1827	0.129	0.1571	0.0224	75	100	75

DBD: Delta Bar Delta

CG: Conjugate-Gradient

QP: Quick Propagation

LM: Levenberg-Marquardt

NN-Model	GFFDNN				GFFDNN-DR			SVM			SVM-DR					
Noise in	In	put	Out	tput	Inp	put	Ou	tput	Inp	put	Out	tput	Inj	put	Out	put
Type of Noise	G	U	G	U	G	U	G	U	G	U	G	U	G	U	G	U
% Variance			Avera	age Cl	assific	ation A	Accura	cy on T	esting	on Te	sting I	Data i.e	. unsee	en Dat	a	
1	75	100	100	50	100	100	100	100	100	100	100	100	100	100	100	100
2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
3	100	100	100	100	100	100	100	100	100	100	100	100	100	100	66.7	100
4	100	100	100	75	100	100	100	100	100	100	100	100	66.7	100	100	100
5	100	100	50	100	100	75	100	100	100	100	100	100	66.7	66.7	100	100
6	100	100	100	100	100	100	0	100	100	100	100	100	100	100	100	100
7	100	100	100	100	100	100	100	100	100	100	100	66.7	100	100	100	100
8	100	75	100	75	100	100	100	100	100	100	100	100	100	100	100	100
9	100	87.5	100	75	75	100	100	100	100	100	100	100	100	100	100	100
10	100	75	100	100	100	75	100	100	100	100	100	100	100	66.7	100	100
11	100	100	100	75	100	87.5	100	100	100	100	100	100	100	100	100	100
12	100	100	100	100	100	75	100	100	100	100	100	100	100	100	100	100
13	100	100	100	100	100	100	100	100	66.7	100	100	100	66.7	66.7	100	100
14	100	100	75	100	100	100	100	100	66.7	100	100	100	100	100	100	100
15	100	100	100	100	100	100	100	87.5	66.7	66.7	66.7	100	66.7	66.7	100	100
16	100	100	100	100	87.5	75	100	100	66.7	66.7	33.3	100	66.7	66.7	100	100
17	100	100	100	100	75	100	100	87.5	66.7	66.7	66.7	100	66.7	66.7	100	66.7
18	100	100	75	75	100	100	100	100	66.7	100	100	66.7	33.3	33.3	100	100
19	100	100	75	75	100	100	62.5	87.5	66.7	66.7	100	66.7	100	100	100	66.7
20	100	87.5	75	100	100	100	100	100	66.7	66.7	100	66.7	33.3	33.3	100	100

Table 3 EFFECT OF NOISE ON AVERAGE CLASSIFICATION ACCURACY WHEN CLASSIFIER TESTED ON TESTING DATA

U: Uniform Noise

G: Gaussian Noise

Z gerformanc Z M e		Testing on T	Fest Data									
	e Performanc	Max. Observed	Min. Observed	Average	SD	Max. Observed	Min. Observed	Average	SD	Т	W	% W
Z	MSE	0.2435	0.00135	0.05273	0.072	0.16856	0.00133	0.03480	0.048			
GFFDN 13-5-4	Percent Correctness	100	75	94.79	9.216	100	83.33	95.42	7.251	1.011	146	40.2
z	MSE	0.28437	0.00022	0.05653	0.077	0.21960	0.00020	0.04624	0.062	0.786		49.3
GFFDNI 5-5-4 O	Percent Correctness	100	75	98.5	5.997	100	80	95.04	7.701		74	
1	MSE	0.84364	0.00529	0.04527	0.120	0.28433	0.01047	0.06055	0.068			
SVM	Percent Correctness	100	83.33	97.41	5.249	100	66.67	97.56	6.842	0.937 472		11.05
1	MSE	0.09926	0.05086	0.05915	0.011	0.094134	0.05454	0.06192	0.007	0.693		44.00
SVM DR	Percent Correctness	100	88.88	99.611	1.944	100	88.88	98.722	3.514		264	

Table 4COMPARATIVE RESULTS

T - Time Elapsed per epoch per exemplar for training in ms.

W – Number of weight connections.

%W – Percent number of weight connections reduced.

N-N Model	Performance	HLTY	BOTH	INT	ECE	Overall
A) A)	MSE	0.0072	0.0023	0.0160	0.0109	0.0091
GFFDN (13-5-	Percent Correctness	100	100	100	100	100
NN (†	MSE	0.0009	0.0719	0.0024	0.0243	0.0249
GFFD DR (5-5-4	Percent Correctness	100	100	100	100	100
Μ	MSE	0.0001	0.0003	0.0002	0.0006	0.0003
SVI	Percent Correctness	100	100	100	100	100
W ~	MSE	0.0036	0.0353	0.0025	0.0294	0.0177
SV] DF	Percent Correctness	100	100	100	100	100

Table 5SAMPLE RESULTS OF NETWORK FOR EACH FAULT

Table 6

SAMPLE DESIRED AND ACTUAL OUTPUT OF NETWORK

	Des	ired		Output of Network							
Η	В	Ι	E	Н	В	Ι	Е				
0	0	0	1	-0.00278	0.009998	-0.01879	1.004026				
0	0	0	1	0.000783	-0.01557	0.00963	0.996425				
0	0	1	0	0.00024	0.00018	1.000255	-0.00071				
0	1	0	0	0.000357	0.999824	0.000434	-0.00062				
0	0	1	0	0.000312	-0.00019	0.999955	-3.1E-05				
0	0	1	0	0.000707	-0.00025	1.000305	-1.2E-05				
0	0	0	1	-0.00139	-0.00034	-0.00027	0.999077				
1	0	0	0	1.009613	0.002908	-0.00255	-0.01547				
0	0	1	0	0.000247	0.001613	0.998812	0.000775				
0	0	1	0	0.000193	-0.00113	1.000305	-0.00061				
0	1	0	0	2.95E-08	0.999843	-5.6E-05	-0.00083				
0	0	0	1	0.001683	-0.00029	-0.00053	1.001546				
0	1	0	0	0.000937	0.997332	0.001464	0.002719				
0	0	1	0	0.000384	-0.00099	1.000301	0.000357				