Computer Assisted Mechanics and Engineering Sciences, 14: 307–316, 2007.

Fault classification in cylinders using multi-layer perceptrons, support vector machines and Gaussian mixture models

Tshilidzi Marwala, Unathi Mahola
School of Electrical and Information Engineering, University of the Witwatersrand Private Bag x 3, Wits 2050, South Africa

Snehashish Chakraverty
Central Building Research Institute, Roorkee 247667, U.A., India

(Received in the final form March 20, 2007)

Gaussian mixture models (GMM) and support vector machines (SVM) are introduced to classify faults in a population of cylindrical shells. The proposed procedures are tested on a population of 20 cylindrical shells and their performance is compared to the procedure, which uses multi-layer perceptrons (MLP). The modal properties extracted from vibration data are reduced into low dimension using the principal component analysis and are then used to train the GMM, SVM and MLP. It is observed that the GMM gives 98% classification accuracy, SVM gives 94% classification accuracy while the MLP gives 88% classification accuracy. Furthermore, GMM is found to be more computationally efficient than MLP which is in turn more computationally efficient than SVM.

1. Introduction

Vibration data have been used with varying degrees of success to classify damage in structures [6]. In the fault classification process, there are various stages involved and these are: data extraction, data processing, data analysis, and fault classification. Data extraction process involves the choice of data to be extracted and the method of extraction. Data that have been used for fault classification include strain concentrations in structures and vibration data where strain gauges and accelerometers are used respectively [6]. In this paper, vibration data processed using modal analysis, are used for fault classification.

In the data processing stage the measured vibration data need to be processed. This is mainly due to the fact that the measured vibration data, which are in the time domain, are difficult to use in raw form. The time-domain vibration data may be transformed into the modal domain, frequency domain and time-frequency domain [16–18]. In this paper, the time-domain vibration data set is transformed into the modal domain where it is represented as natural frequencies and mode shapes.

The data processed need to be analysed and the general trend has been to automate the analysis process and thus automate the fault classification process. To achieve this goal intelligent pattern recognition process needs to be employed and methods such as neural networks have been widely applied [16–18]. There are many types of neural networks that have been employed and these include multi-layer perceptron (MLP), radial basis function (RBF) and Bayesian neural networks [2, 3]. Recently, new pattern recognition methods called support vector machines (SVMs) [13] and Gaussian mixture models (GMMs) [19, 22] have been proposed and found to be particularly suited to classification problems. SVMs have been found to outperform neural networks [23]. One of the examples where the fault classification process summarized at the beginning of this paper has been implemented is fault classification in a population of nominally identical cylindrical shells [16–18].

The importance of fault identification process in a population of nominally identical structures is particularly important in areas such as automated manufacturing process in the assembly line. Thus far various forms of neural networks such as MLP and Bayesian neural networks have been successfully used to classify faults in structures [12]. Worden and Lane [27] used SVMs to identify damage in structures. However, SVMs have not been used for fault classification in a population of cylinders. Based on the successes of SVMs and GMMs observed in other areas [22, 23], we therefore propose in this paper SVMs and GMMs for classifying faults in a population of nominally identical cylindrical shells.

This paper is organized as follows: MLPs, SVMs and GMMs are summarized, experiment performed is discussed and results as well as conclusions are discussed.

2. Multi-layer perceptron (MLP) neural networks

MLP neural networks are parameterised graphs that make probabilistic assumptions about data and in this paper these data are modal domain data and their respective classes of faults. In this paper, MLP neural networks are trained to give a relationship between modal domain data and the fault classes. As mentioned earlier, there are several types of neural network procedures such as multi-layer perceptron, radial basis function, Bayesian networks and recurrent networks [2, 3] and in this paper the MLP is used.

This network architecture contains hidden units and output units and has one hidden layer. For the MLP the relationship between the output y, representing fault class, and x, representing modal data, may be may be written as follows [2, 3, 10, 11]:

$$y_k = f_{\text{outer}}\left(\sum_{j=1}^{M} w_{kj}^{(2)} f_{\text{inner}}\left(\sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right). \tag{1}$$

Here, $w_{ji}^{(1)}$ and $w_{ji}^{(2)}$ indicate weights in the first and second layer, respectively, going from input i to hidden unit j, M is the number of hidden units, d is the number of input units while $w_{j0}^{(1)}$ and $w_{k0}^{(2)}$ indicate the biases for the hidden unit j and the output unit k. In this paper, the function $f_{\text{outer}}(\cdot)$ is logistic while f_{inner} is a hyperbolic tangent function. Training the neural network identifies the weights in Eqs. (1) and in this paper the scaled conjugate gradient method is used [20]. The weights and biases in the hidden layers are varied until the error between the network prediction and the output from the training data is minimized. A cost function must be chosen in order to apply an optimization technique. A cost function is a mathematical representation of the overall objective of the problem and in this section the following cost function is used [3],

$$E = \sum_{n=1}^{N}\sum_{k=1}^{K}\{t_{nk} - y_{nk}\}^2 + \frac{\alpha}{2}\sum_{j=1}^{W} w_j^2 . \tag{2}$$

In Eq. (2), n is the index for the training pattern, k is the index for the output units, N is the number of training patterns, K is the number of output units and W is the number of weights. The sum-of-square-of-errors cost function, which is the first term in Eq. (2), tends to give similar absolute error for each pattern. As a result, it performs poorly on target values of small magnitude. The second term in Eq. (2) is the regularization parameter and it penalizes weights of large magnitudes [26]. This regularization parameter is called the weight decay and its coefficient, α, determines the relative contribution of the regularization term to then training error. This regularization parameter ensures that the mapping function is smooth. Including the regularization parameter has been found to give significant improvements in network generalization [11]. In neural networks, to produce an over-fitted mapping with regions of large curvature requires large weights. The weight decay regularization penalizes large weights thereby encouraging the weights to be small and avoiding an over-fitted

mapping between the inputs and the outputs. If α is too high then the regularization parameter over-smoothens the network weights giving inaccurate results. If α is too small then the effect of the regularization parameter is negligible and unless other measures that control the complexity of the model, such as the early stopping method [2], are implemented then the trained network becomes too complex and thus performs poorly on the validation set. Before minimization of the cost function is performed, the network architecture needs to be constructed by choosing the number of hidden units, M. If M is too small, the neural network will be insufficiently flexible and will give poor generalization of the data because of high bias. However, if M is too large, the neural network will be unnecessarily flexible and will give poor generalization due to a phenomenon known as over-fitting caused by high variance [8].

3. Support vector machines (SVMs)

According to [21], the classification problem can be formally stated as estimating a function $f : RN \to \{-1, 1\}$ based on an input-output training data generated from an independently, identically distributed unknown probability distribution $P(x, y)$ such that f will be able to classify previously unseen (x, y) pairs. Here x is the modal data while y is the fault class. The best such function is the one that minimizes the expected error (risk) which is given by [21]

$$R[f] = \int l(f(x), y)\, \mathrm{d}P(x, y) \tag{3}$$

where l represents a loss function. Since the underlying probability distribution P is unknown, Eq. (3) cannot be solved directly. The best we can do is to find an upper bound for the risk function [26] that is given by

$$R[f] = R[f]_{\text{emp}} + \sqrt{\frac{h\left(\ln \frac{2n}{h} + 1\right) - \ln \frac{\delta}{4}}{n}} \tag{4}$$

where $h \in N^+$ is the Vapnik–Chervonenkis (VC) dimension of $f \in F$ and $\delta > 0$ holds true for all δ. The VC dimension of a function class F is defined as the largest number h of points that can be separated in all possible ways using functions of the class [26]. The empirical error $R[f]_{\text{emp}}$ is a training error given by [26]

$$R[f]_{\text{emp}} = \frac{1}{n} \sum_{i+1}^{n} l(f(x_i), y_i)\,. \tag{5}$$

Assuming that the training sample is linearly separable by a hyperplane of the form

$$f(x) = \langle w, x \rangle + b \qquad \text{with} \quad w \in \chi, \quad b \in \Re \tag{6}$$

where w is an adjustable weight vector and b is an offset, the classification problem looks like in Fig. 1.

The goal of the learning algorithm as proposed by [26] is to find the hyperplane with maximum margin of separation from the class of separating hyperplanes. But since real-world data often exhibit complex properties, which cannot be separated linearly, more complex classifiers are required. In order to avoid the complexity of the nonlinear classifiers, the idea of linear classifiers in a feature space comes into place. Support vector machines try to find a linear separating hyperplane by first mapping the input space into a higher dimensional feature space F. This implies each training example x_i is substituted with $\Phi(x_i)$ [26],

$$y_i(w \cdot \phi(x_i) + b), \qquad i = 1, 2, \ldots, n. \tag{7}$$

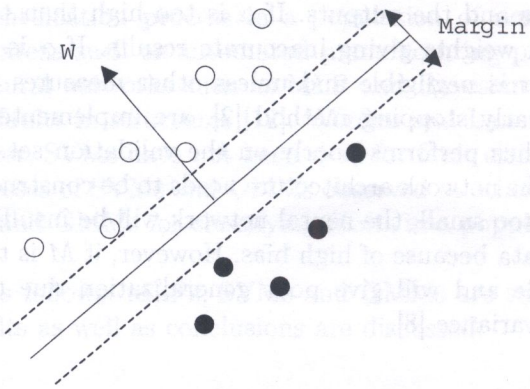

Fig. 1. A linear SVM classifier and margins: a linear classifier is defined by a hyperplane's normal vector w and an offset b, i.e. the decision boundary is $\{x|w \cdot x + b = 0\}$ (thick line); each of the two half spaces defined by this hyperplane corresponds to one class, i.e. $f(x) = \text{sign}((w \cdot x) + b)$ [26]

The VC dimension h in the feature space F is bounded according to $h \leq \|W\|^2 R^2 + 1$ where R is the radius of the smallest sphere around the training data [26]. Hence minimizing the expected risk is stated as an optimization problem [25],

$$\min_{w,b} \ \frac{1}{2}\|w\|^2. \tag{8}$$

However, assuming that we can only access the feature space using only dot products, Eq. (8) is transformed into a dual optimization problem by introducing Lagrangian multipliers α_i, $i = 1, 2, \ldots, n$ and doing some minimization, maximization and saddle point property of the optimal point [4, 9, 26] the problem becomes

$$\begin{aligned} &\max_{w,b} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i\,,\, x_j) \\ &\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \ldots, n, \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{9}$$

The Lagrangian coefficients, α_i, are obtained by solving Eq. (9) which in turn is used to solve w to give the non-linear decision function [25, 26],

$$f(x) = \text{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i \left(\Phi(x) \cdot \Phi(x_i)\right) + b\right) = \text{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i k(x, x_i) + b\right). \tag{10}$$

In the case when the data is not linearly separable, a slack variable ξ_i, $i = 1, \ldots, n$, is introduced to relax the constraints of the margin as

$$y_i((w \cdot \Phi(x_i)) + b) \geq 1 - \zeta_i\,, \qquad \zeta_i \geq 0, \quad i = 1, \ldots, n. \tag{11}$$

A trade-off is made between the VC dimension and the complexity term of Eq. (4), which gives the optimization problem

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} \xi_i \tag{12}$$

where $C > 0$ is a regularization constant that determines the above-mentioned trade-off. The dual optimization problem is then given by [25, 26],

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$
$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n, \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{13}$$

A Karush–Kuhn–Tucker (KKT) condition which says that only the α_i associated with the training values x_i on or inside the margin area have non-zero values, is applied to the optimization problem in Eq. (13) to find the α_i, the threshold variable b and the decision function f [25].

4. Gaussian mixture models (GMMs)

Gaussian mixture models have been a reliable classification tool in many applications of pattern recognition such as speech and face recognition. The success of GMM in classification of dynamic signals has also been observed by many researchers who compared GMM and MLP and found that the GMM approach easily outperforms the MLP approach for high resolution faces and is significantly more robust in imperfectly located faces [22]. These observations showed that the computational requirements of the GMM approach can be significantly smaller than the MLP approach. Due to the dynamic nature of vibration signals, GMM seem very suitable for classification of cylinder faults. Like in text-independent speech signals, there is no prior knowledge on how the vibration signal will be when a fault exists. Other advantages of using a GMM are that it is computationally inexpensive and is based on well understood statistical models [24]. The GMM works by creating a maximum likelihood model for each fault case given by [5, 24]

$$\lambda = \{w, \mu, \Sigma\}, \tag{14}$$

where, w, μ, Σ are the weights, means and diagonal covariance of the features. Given a collection of training vectors the parameters of this model are estimated by a number of algorithms such as the Expectation-Maximization (EM) algorithm [5]. In this paper, the EM algorithm is used since it has reasonably fast computational time when compared to other algorithms. The EM algorithm finds the optimum model parameters by iteratively refining GMM parameters to increase the likelihood of the estimated model for the given fault feature modal vector. For the EM equations for training a GMM, the reader is referred to [5]. Fault detection or diagnosis using this classifier is then achieved by computing the likelihood of the unknown modal data of the different fault models. This likelihood is given by [5]

$$\hat{s} = \arg \max_{1 \leq f \leq F} \sum_{k=1}^{K} \log p(x_k | \lambda_f) \tag{15}$$

where F represents the number of faults to be diagonalized, $X = \{x_1, x_2, \ldots, x_K\}$ is the unknown D-dimension fault modal data and $p(x_k|\lambda_f)$ is the mixture density function given by [5],

$$p(x|\lambda) = \sum_{i=1}^{M} w_i p(x). \tag{16}$$

By assuming Gaussian probability density function then

$$p_i(x_t) = \frac{1}{(2\pi)^{D/2} \sqrt{\Sigma_i}} \exp\left\{ -\frac{1}{2} (x_k - \mu_i)^T (\Sigma_i)^{-1} (x_k - \mu_i) \right\}. \tag{17}$$

It should be noted that the mixture weights, w_i, satisfy the constraints $\sum_{i=1}^{M} w_i = 1$.

The training of the GMM is a fast and straightforward process which estimates the mean and covariance parameters from the training data [19, 24]. The training procedure estimates the model parameters from a set of observations using the EM algorithm. The EM algorithm tries to increase the expected log-likelihood of the complete data x given the partially observed data and finds the optimum model parameters by iteratively refining GMM parameters for a given cylinder fault feature vector. In this paper, only the diagonal covariance matrices are used. The reason for this is that the diagonal covariance matrix was found to be more computationally efficient and outperforms the full covariance matrix.

5. Input data

This section describes the inputs that are used to test the SVM, MLP and GMM. When modal analysis is used for fault classification it is often found that there are more parameters extracted from the vibration data than can be possibly used for MLP, SVM and GMM training. These data must therefore be reduced because of a phenomenon called the curse of dimensionality [1], which refers to the difficulties associated with the feasibility of density estimation in many dimensions. However, this reduction process must be conducted such that the loss of essential information is minimized. The techniques implemented in this paper to reduce the dimension of the input data remove parts of the data that do not contribute significantly to the dynamics of the system being analysed or those that are too sensitive to irrelevant parameters. To achieve this, we implement the principal component analysis [14], which is discussed in the next section.

5.1. Principal component analysis

In this paper we use the principal component analysis (PCA) [14] to reduce the input data into independent input data. The PCA orthogonalizes the components of the input vector so that they are uncorrelated with each other. In the PCA, correlations and interactions among variables in the data are summarised in terms of a small number of underlying factors.

6. Foundations of dynamics

As indicated earlier, in this paper modal properties i.e. natural frequencies and mode shapes are extracted from the measured vibration data and used for fault classification. For this reason the foundation of these parameters are described in this section. All elastic structures may be described the time domain as [7, 15]

$$[M]\{X''\} + [C]\{X'\} + [K]\{X\} = \{F\} \tag{18}$$

where $[M]$, $[C]$ and $[K]$ are the mass, damping and stiffness matrices respectively, and $\{X\}$, $\{X'\}$ and $\{X''\}$ are the displacement, velocity and acceleration vectors, respectively, while $\{F\}$ is the applied force vector. If Eq. (18) is transformed into the modal domain to form an eigenvalue equation for the i-th mode, then [7, 15]

$$\left(-\overline{\omega}_i^2[M] + \mathrm{j}\overline{\omega}_i[C] + [K]\right)\{\overline{\phi}\}_i = \{0\} \tag{19}$$

where $\mathrm{j} = \sqrt{-1}$, $\overline{\omega}_i$ is the i-th complex eigenvalue, with its imaginary part corresponding to the natural frequency ω_i, $\{0\}$ is the null vector, and $\{\overline{\phi}\}_i$ is the i-th complex mode shape vector with the real part corresponding to the normalized mode shape $\{\phi\}_i$. From Eq. (19) it is evident that changes in the mass and stiffness matrices cause changes in the modal properties and thus modal properties are damage indicators.

7. EXAMPLE: CYLINDRICAL SHELLS

7.1. Experimental procedure

In this section the procedures of using GMM and SVM are experimentally validated and compared to the procedure of using MLP. The experiment is performed on a population of cylinders, which are supported by inserting a sponge rested on a bubble-wrap, to simulate a 'free–free' environment (see Fig. 2). The impulse hammer test is performed on each of the 20 steel seam-welded cylindrical shells. The impulse is applied at 19 different locations as indicated in Fig. 2. When linearity tests were performed it was found that the structure remains linear and more details on this experiment may be found in [18].

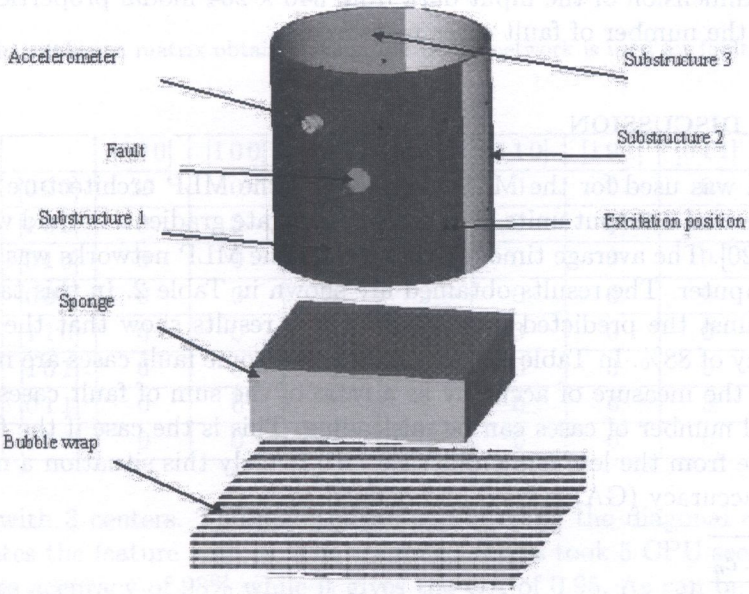

Fig. 2. Illustration of a cylindrical shell showing the positions of the impulse, accelerometer, substructures, fault position and supporting sponge

Each cylinder is divided into three equal substructures and holes of 10–15 mm in diameter are introduced at the centers of the substructures to simulate faults.

For one cylinder the first type of fault is a zero-fault scenario. This type of fault is given the identity [0 0 0], indicating that there are no faults in any of the three substructures. The second type of fault is a one-fault-scenario, where a hole may be located in any of the three substructures. Three possible one-fault-scenarios are [1 0 0], [0 1 0], and [0 0 1] indicating one hole in substructures 1, 2 or 3, respectively. The third type of fault is a two-fault scenario, where holes are located in two of the three substructures. Three possible two-fault-scenarios are [1 1 0], [1 0 1], and [0 1 1]. The final type of fault is a three-fault-scenario, where a hole is located in all three substructures, and the identity of this fault is [1 1 1]. There are 8 different types of fault-cases considered (including [0 0 0]).

Each cylinder is measured three times under different boundary conditions by changing the orientation of a rectangular sponge inserted inside the cylinder. The number of sets of measurements taken for undamaged population is 60 (20 cylinders $\times$ 3 for different boundary conditions). In the 8 possible fault types, two fault types [0 0 0] and [1 1 1] has 60 number of occurrences while the rest has 24. It should be noted that the numbers of one- and two-fault cases are each 72. This is because as mentioned above, increasing the sizes of holes in the substructures and taking vibration measurements generated additional one- and two-fault cases. Fault cases used to train and test the networks are shown in Table 1.

Table 1. Fault cases used to train, cross-validates and test the networks; the multifold cross-validation technique is used because of the lack of availability of the data

Fault	[0 0 0]	[1 0 0]	[0 1 0]	[0 0 1]	[1 1 0]	[1 0 1]	[0 1 1]	[1 1 1]
Training set	21	21	21	21	21	21	21	21
Test set	39	3	3	3	3	3	3	39

The impulse and response data are processed using the Fast Fourier Transform (FFT) to convert the time domain impulse history and response data into the frequency domain. The data in the frequency domain are used to calculate the frequency response functions (FRFs). From the FRFs, the modal properties are extracted using modal analysis [7]. The number of modal properties identified is 340 (17 modes $\times$ 19 measured mode-shape-co-ordinates + 17 natural frequencies). The PCA are used to reduce the dimension of the input data from 340 $\times$ 264 modal properties to 10 $\times$ 264. Here 264 corresponds to the number of fault cases measured.

8. Results and discussion

The measured data was used for the MLP training and the MLP architecture contained 10 input units, 8 hidden units and 3 output units. The scaled conjugate gradient method was used for training the MLP network [20]. The average time it took to train the MLP networks was 12 CPU seconds on the Pentium II computer. The results obtained are shown in Table 2. In this table the actual fault cases are listed against the predicted fault cases. These results show that the MLP classify fault cases to the accuracy of 88%. In Table 1 it was shown that some fault cases are more numerous than others. In this case the measure of accuracy as a ratio of the sum of fault cases classified correctly divided by the total number of cases can be misleading. This is the case if the fault cases classified incorrectly are those from the less numerous cases. To remedy this situation a measure of accuracy called geometrical accuracy (GA) is used and defined as

$$GA = \sqrt{\frac{\prod c_1 \dots c_n}{\prod q_1 \dots q_n}} \tag{20}$$

where $\prod$ is the product, c_n is the number of n-th fault cases classified correctly while q_n is the n-th fault class. Using this measure the MLP gives the accuracy of 0.7. On training the SVM, there are different parameters that can be changed namely the capacity and the e-insensitivity, the amount of training inputs and the function to be used for the kernel. Some of the functions that can be used are: linear, radial basis function, sigmoid, and spline. In this paper the exponential radial basis function is used as a kernel. The training process took 45 CPU seconds and the capacity was set to infinity. The results obtained are shown in Table 3 and these results show that the SVM gives accuracy of 94% while it gives the GA of 0.92. GMM architecture on the other hand used diagonal

Table 2. The confusion matrix obtained when the MLP network is used for fault classification

		Predicted							
		[0 0 0]	[1 0 0]	[0 1 0]	[0 0 1]	[1 1 0]	[1 0 1]	[0 1 1]	[1 1 1]
Actual	[0 0 0]	39	0	0	0	0	0	0	0
	[1 0 0]	0	3	0	0	0	0	0	0
	[0 1 0]	0	0	3	0	0	0	0	0
	[0 0 1]	0	0	0	3	0	0	0	6
	[1 1 0]	0	0	0	0	3	1	0	0
	[1 0 1]	0	0	0	0	0	2	0	0
	[0 1 1]	0	0	0	0	0	0	3	4
	[1 1 1]	0	0	0	0	0	0	0	29

Table 3. The confusion matrix obtained when the SVM network is used for fault classification

		Predicted							
		[000]	[100]	[010]	[001]	[110]	[101]	[011]	[111]
Actual	[000]	39	0	0	0	0	0	0	0
	[100]	0	3	0	0	0	0	0	0
	[010]	0	0	3	0	0	0	0	0
	[001]	0	0	0	3	0	0	0	5
	[110]	0	0	0	0	3	0	0	0
	[101]	0	0	0	0	0	3	0	0
	[011]	0	0	0	0	0	0	3	1
	[111]	0	0	0	0	0	0	0	33

Table 4. The confusion matrix obtained when the GMM network is used for fault classification

		Predicted							
		[000]	[100]	[010]	[001]	[110]	[101]	[011]	[111]
Actual	[000]	39	0	0	0	0	0	0	0
	[100]	0	3	0	0	0	0	0	0
	[010]	0	0	3	0	0	0	0	0
	[001]	0	0	0	3	0	0	0	1
	[110]	0	0	0	0	3	0	0	1
	[101]	0	0	0	0	0	3	0	0
	[011]	0	0	0	0	0	0	3	2
	[111]	0	0	0	0	0	0	0	35

covariance matrix with 3 centers. The main advantage of using the diagonal covariance matrix is that this de-correlates the feature vectors. The training process took 5 CPU seconds. Table 4 shows that the GMM gives accuracy of 98% while it gives the GA of 0.95. As can be seen from Tables 2, 3 and 4, the GMM outperforms the SVM network, which out-performed the MLP.

9. Conclusions

In this paper GMM and SVM were introduced to classify faults in a population of cylindrical shells and compared to the MLP. The modal properties were extracted from vibration data and then reduced into low dimension using the principal component analysis. When the parameters in reduced dimension are used to train the GMM, SVM and MLP, it is observed that the GMM produces 98% classification accuracy, SVM produces 94% classification accuracy while the MLP produces 88% classification accuracy rates. It is therefore concluded that the GMM gives more accurate results than the SVM, which in turn gives more accurate results than the MLP. Furthermore, GMMs are found to be more computationally efficient than MLPs which are in turn more computationally efficient than SVMs.

References

[1] R. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, New Jersey, USA, 1961.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, UK, 1995.

[3] C.B. Bishop. *Pattern Recognition and Machine Intelligence.* Springer, Berlin, 2006.

[4] C.A. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**: 121–167, 1998.

[5] A. Dempster, N. Laird, D. Rubin. Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, **39**: 1–38, 1977.

[6] S.W. Doebling, C.R. Farrar, M.B. Prime, D.W. Shevitz. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. *Los Alamos Technical Report LA-13070-MS*, Los Alamos National Laboratory, New Mexico, USA, 1996.

[7] D.J. Ewins. *Modal Testing: Theory and Practice*, Research Studies Press, Letchworth, UK, 1995.

[8] S. Geman, E. Bienenstock, R.R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, **4**: 1–58, 1992.

[9] E. Habtemariam, T. Marwala, M. Lagazio. Artificial intelligence for conflict management. *Proceedings of the IEEE International Joint Conference on Neural Networks*, Montreal, Canada, pp. 2583–2588, 2005.

[10] S. Haykin. *Neural networks*. Prentice-Hall, New York, USA, 1995.

[11] G.E. Hinton, Learning translation invariant recognition in massively parallel networks. In: J.W. de Bakker, A.J. Nijman, P.C. Treleaven, eds., *Proceedings PARLE Conference on Parallel Architectures and Languages Europe*, pp. 1–13. Springer-Verlag, Berlin, 1987.

[12] A. Iwasaki, A Todoroki, Y. Shimamura, *et al.*. An unsupervised statistical damage detection method for structural health monitoring (applied to detection of delamination of a composite beam). *Smart Materials and Structures*, **13**: 80–85, 2004.

[13] J. Joachims. Making large-scale SVM learning practical. In: B. Schölkopf, C.J.C. Burges, A.J. Smola, eds., *Advances in Kernel Methods – Support Vector Learning*, pp. 169–184. MIT Press, Cambridge, MA, 1999.

[14] I.T. Jollife. *Principal Component Analysis*. Springer-Verlag , New York, USA, 1986.

[15] N.M.M. Maia, J.M.M. Silva. *Theoretical and Experimental Modal Analysis*. Research Studies Press, Letchworth, U.K, 1997.

[16] T. Marwala. Fault identification using pseudomodal energies and modal properties. *AIAA Journal*, **39**: 1608–1618, 2001.

[17] T. Marwala. Probabilistic fault identification using a committee of neural networks and vibration data. *AIAA, Journal of Aircraft*, **38**: 138–146, 2001.

[18] T. Marwala. *Fault Identification Using Neural Networks and Vibration Data*. University of Cambridge Ph.D. Thesis, Cambridge, UK, 2001.

[19] L.M. Meng, G. Frei, I. Osorio, G. Strang, T.Q. Nguyen. Gaussian mixture models of ECoG signal features for improved detection of epileptic seizures, *Medical Engineering and Physics*, **26**: 379–393, 2004.

[20] M. Møller. A scaled conjugate gradient algorithm for fast-supervised learning. *Neural Networks*, **6**: 525–533, 1993.

[21] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, **12**: 181–201, 2001.

[22] F.V. Nelwamondo, T. Marwala, U. Mahola. Early classifications of bearing faults using hidden Markov models, Gaussian mixture models, Mel-frequency cepstral coefficients and fractals. *International Journal of Innovative Computing, Information and Control*, **2**: 1281–1299, 2006.

[23] M.M. Pires, T. Marwala. American option pricing using multi-layer perceptron and support vector machines. *Proceedings of the IEEE Conference in Systems, Man and Cybernetics*, The Hague, pp. 1279–1285, 2004.

[24] D.A. Reynolds, T.F. Quatieri, R.B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, **10**: 19–41, 2000.

[25] B. Schölkopf, A.J. Smola. A short introduction to learning with kernels. *Proceedings of the Machine Learning Summer School*, pp. 41–64. Springer, Berlin, 2003.

[26] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, Berlin, 1995.

[27] K. Worden, A.J. Lane. Damage identification using support vector machines. *Smart Materials and Structures*, **10**: 540–547, 2001