International Journal of the Physical Sciences Vol. 6(13), pp. 3181-3188, 4 July, 2011
Available online at http://www.academicjournals.org/IJP
DOI: 10.5897/AJBM09.09
ISSN 1992 - 1950 ©2011 Academic Journa

Full Length Research Paper

Comparison of Naïve bayes classifier with back propagation neural network classifier based on *f* - folds feature extraction algorithm for ball bearing fault diagnostic system

O. Addin[1], S. M. Sapuan[2]*, M. Othman[3] and B. A. Ahmed Ali[4]

[1]Laboratory of Intelligent Systems, Institute of Advanced Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.
[2]Department of Mechanical and Manufacturing Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.
[3]Department of Communication Technology and Networks, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.
[4]Institute of Advanced Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.

Accepted 18 April, 2011

This paper is intended to compare the Naïve bayes classifier for ball bearing fault diagnostic system with the back propagation neural network based on the *f*-folds feature extraction algorithm. The *f*-folds feature extraction algorithm has been used with different number of folders and clusters. The two classifiers have shown similar classification accuracies. The Naive bayes classifier has not shown any case of false negative or false positive classification. However, the back propagation neural network classifier has shown many cases of false positive and false negative classifications.

Key words: Neural network classifier, Naive bayes classifier, diagnostic system, engineering materials.

INTRODUCTION

Most moving machinery parts are bedded in supporting elements (for example, sliding and rolling ball bearings). Rolling contact is a phenomenon that occurs in many applications of precision-machined components (for example, bearings and shafts) in automotive, aircraft, aerospace, and other industries (Guo and Dale, 2005). The main task of all types of bearings is to minimize energetic loss and ensure maximum lifetime of bedding. Lifetime of rolling bearings means the period in which the bearing carries its function until it cannot meet the requirements of operation and has to be put out of service. In practical situations, ball bearing failure or damage may occur during manufacturing processes or in-service. Contact damage may happen in the bearings, which is caused by cyclically repeating processes in surface layer of material by mutual dynamic load of two bodies. Damage of surface layers causes inception of micro-cracks in places of maximum sheer stress, by progressive separating of damaged surface layers and by inception of holes on the surface (Figure 1). In the beginning, this fatigue damage results in decrease of functional properties of damaged part; however, emerged surface hole may gradually create a centre of fatigue crack, which successively enlarges to the whole section of the part. In some applications (for example, airplanes and space shuttles), damages of these bearing elements have the potential of growing and leading to considerable material loss and particularly it might cause catastrophic loss of human life, and economical loss (Kessler et al., 2002).

Health monitoring and online damage detection engineering materials is of growing importance in many fields. With the increasing demand of safe space technology, the various structural systems that compose

*Corresponding author. E-mail: sapuan@eng.upm.edu.my

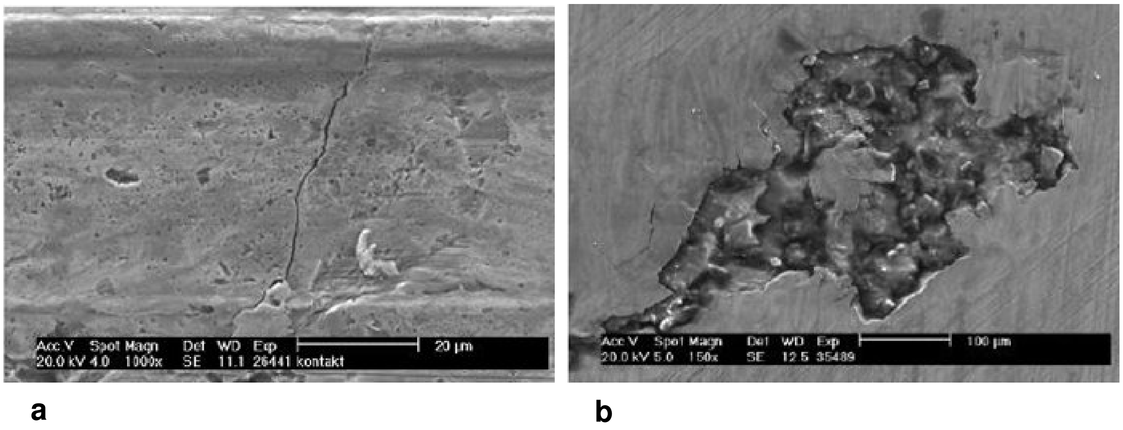

Acc.V Spot Magn Det WD Exp 20 µm
20.0 kV 4.0 1000x SE 11.1 26441 kontakt
Acc.V Spot Magn Det WD Exp 100 µm
20.0 kV 5.0 150x SE 12.5 35489
a
b

Figure 1. Micrographs (Kessler et al., 2002) of short crack in contact fatigue trace cast iron (a) and developed pitting in carbon steel (b).

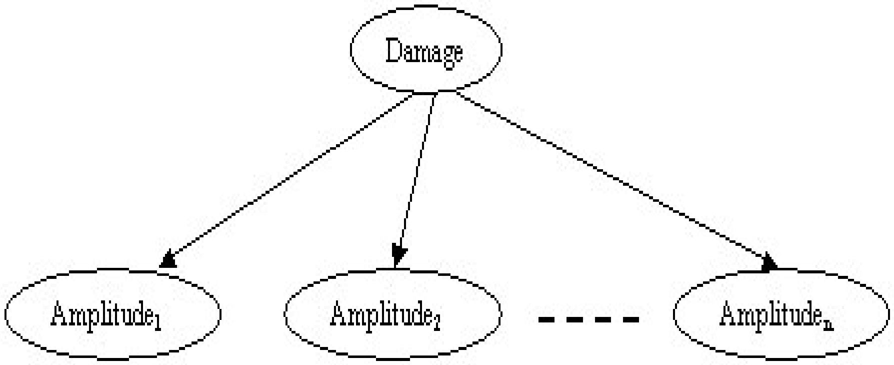


Figure 2. A Naïve bayes for damage detection by using some amplitudes of wave.

air and space vehicles must be monitored for safety and reliability. Hence, the current most common methods of visual inspection and time-based maintenance will be upgraded to online monitoring of the integrity of the vehicle and conditioned-based maintenance (D'Souza and Epureanu, 2005).

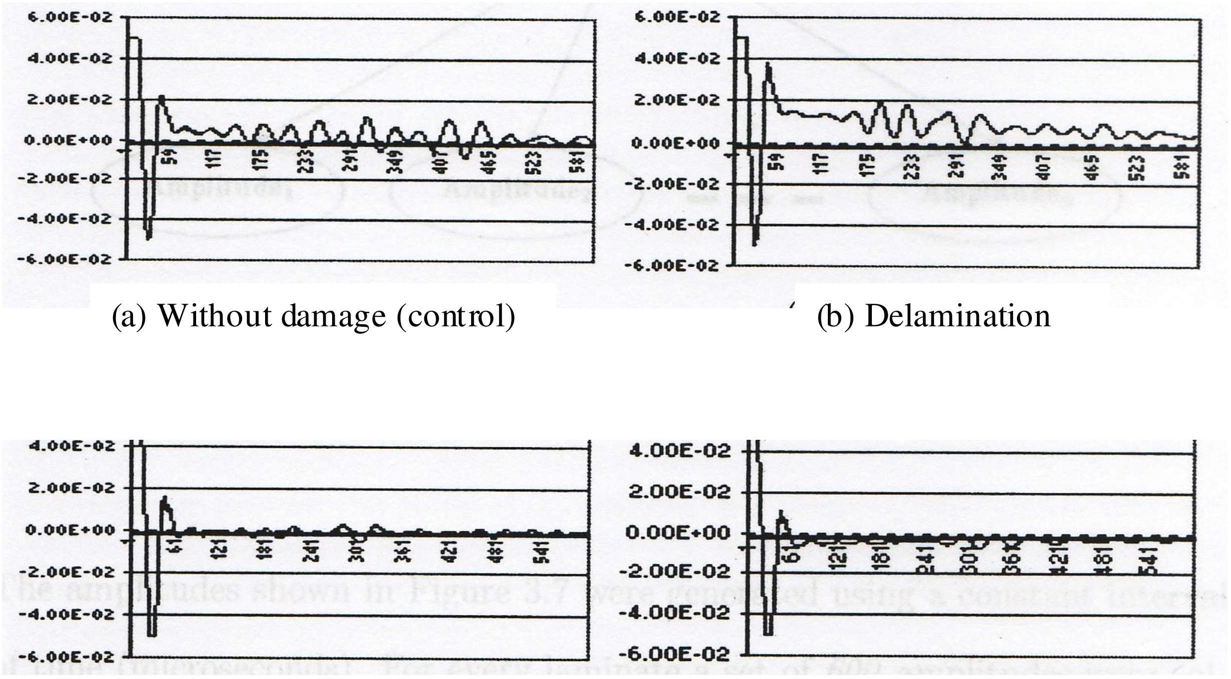
Classification is a basic task in damage detection that requires the construction of a classifier that is a function, which assigns a class label to instances described by a set of attributes. Numerous approaches to this problem are based on various functional representations such as neural networks. The automation of damage detection needs to be able to take into account the affect of a wide range of possible actions on a large number of factors that are linked together. The problem is to find a technique that can take into account all these factors without declining the efficiency of the damage detection. One way is to use Naïve bayes, a type of model-based decision support system already used successfully in many fields, for example, artificial intelligence (D'Souza and Epureanu, 2005; Duda and Hart, 1973).

NAÏVE BAYES CLASSIFIER

Naïve bayes has a strong assumption that all variables in the network are independent of the classification variable (Figure 2). It is very easy to build a Naïve bayes network structure, and it does not require a structured learning algorithm.

The Naïve bayes classifier has several properties that makes it surprisingly useful in practice, despite the fact that the far-reaching independence assumptions are often violated. Like all probabilistic classifier under the MAP decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class, class probabilities do not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying naïve probability model.

The Naïve bayes has two main advantages over other classifiers. First, it is easy to construct and no learning procedure is required. Secondly, the classification process is very efficient since it assumes that all the features



(a) Without damage (control)

(b) Delamination

(c) Crack

(d) Hole

Figure 3. Time trace of amplitudes from graphite/epoxy laminates (Kessler et al., 2002).

are independent of each other. In practical classification problems, it is hardly to come across a situation where the variables are truly conditionally independent of each other. Nevertheless, the Naïve bayes classifier outperformed many sophisticated classifiers on data sets where the variables are not strongly correlated.

The Naïve bayes classifier learns from training data the conditional probability of each variable X_i given the class label *C*. The classification is then done by applying Bayes rule to calculate the probability of *C* given the particular instance of $X_1, X_2, \ldots .. X_n$, and then predicting the class with the highest posterior probability as given.

$$P(C_i|X) = P(C_i)P(X|C_i)/P(X)\_= P(Ci)\prod^{N}_{j=1} P(x_j|C_i)/\sum^{k=1}_{K} P(C_k))\prod^{N}_{j=1} P(x_j|C_k)$$

Where *K* is the number of classes, *J* is the number of variables, and $P(x_j|C_k)$ is the conditional probability for the observed value of variable j given the class C_k. The product of conditional probabilities comes from the assumption that variables are independent given the class, which greatly simplifies the computation of the class scores and eases the induction process. After calculating $P(C_i|X)$ for each class, the algorithm assigns the instance to the class with the highest overall score or probability.

Although the aforeseen formulation of Naïve bayes is the traditional one, we can express the score for each class in another form that is more tractable for analytical purpose. The basic idea is that, if we are concerned only with predictive accuracy, we can invoke any monotonic transformation that does not affect the ordering on class scores. One transformation involves removing the denominator, which is the same for each class, and another involves taking the logarithm for the numerator. Together, these produced a new formula as:

$$S_c = \log P(C) + \sum \log P(x_j|C)$$

The amplitudes shown in Figure 3 represent voltage amplitudes of Lamb-waves produced and collected by piezoelectric transducer (PZT) sensors and actuators mounted on the surface of quasi-isotropic graphite/epoxy laminates. The first specimen is a control unit (laminate without damage), and the rest of the specimen contain artificial damages. These damages are delimitation, crack, and hole. The figure shows sound waves behave differently when passing through the laminate without and with damage, and every damage produce different amplitudes. Amplitudes with many cases and different kind of damages can be used to learn the conditional probability tables of variables *(P (Amplitude | Damage))* in the network. Ultimately, the model can be used to predict the damages in laminated composite materials with the highest posterior probability. The probabilities of the damages are determined by entering the new evidence obtained from the amplitudes of the new case to the network.

The amplitudes shown in Figure 3 were generated using a constant interval of time (microseconds). For every laminate a set of 600 amplitudes were collected. If all of these amplitudes were used as variables on the damage detection model, the model would be overwhelmed, complicated, and its accuracy might slightly be decreased. Different techniques have been adopted for feature subset selections to decrease the size of the data and increase the accuracy.

These methods can be divided into two types, feature selection and feature extraction. In feature selection, the integrity of the original features is preserved. But it costs a great degree of time complexity for an exhaustive comparison if a large number of features is to be selected. In contrast, feature extraction is considered as a process to generate a new and smaller feature set by combining the original features. There are various feature extractions available like Principal component analysis (PCA), Independent component analysis (ICA), zone based hybrid feature extraction, etc., they have wide range of applications in different types of classifications, such as text classification, DNA micro-array data analysis, image recognition, image retrieval and so on.

Some of these techniques extract the peaks of the amplitudes as feature subsets, but it is very difficult to be sure whether these peaks can be representative to the whole wave. The rest of the techniques have different kinds of limitations and disadvantages. So as to overcome some of these limitations and tackle some of these disadvantages, the *f*-folds feature subset extraction algorithm (*f*-FFE) has been developed. By introducing this feature extraction algorithm, selecting a suitable tool for the classifiers, implementing and evaluating the extracted features in the classifier. The *f*-FFE method is implemented on the data set to extract features (form new data set), believed to minimize the data set and increase the accuracy of the Naïve bayes classifier.

f – FOLDS FEATURE EXTRACTION ALGORITHM (f-FFE)

In Figure 3, the amplitudes formed using a constant interval of time (microseconds). A different data set might be acquired, if the interval value had been changed. If it had been assumed that the interval was increased 10 times more than the original one, then the original amplitudes would be divided into 60 folds (10 amplitudes in each fold). In this case 10 different data sets would be formed each with 60 amplitudes. The amplitudes included in each set depend on the first amplitude selected from the first fold, if the first amplitude was the first to be included, then the first amplitudes in other folds would be included to the data set, if the second one was the first one to be included, then the seconds in all other folds would be included in the data set etc. This has been used as a base to formalize the *k*-folds feature subset selection algorithm shown subsequently.

Algorithm 1 (f - folds feature selection algorithm)
Input:

Amps = *amp*$_1$, *amp*$_2$,… , *amp*$_n$ (Amplitudes to be clustered). k (number of clusters), f (number of folds).

Outputs:

Means = {*m*(c_1), *m*(c_2), … , *m*(c_k) }
Maxs = {*max*(c_1), *max*(c_2), …, *max*(c_k) }
Mins = {*min*(c1), *min*(c_2), …, *min*(c_k)}

Procedure clustering

```
1. Divide Amps into f folds (fold(1), fold(2),… , fold(f)), where
2. $|fold(1)| = |fold(2)| = … = | fold(f)|$, $fold(i) = \{fold(j)_1, fold(j)_2,…, fold(j)_m\}$, $m = n / f$ and $1 ≤ j ≤ f$.
3. Create a new data set $NewAmp$ ={ $nAmp(1)$, $nAmp(2)$,… , $nAmp$(m)} where $\forall A = fold(k)_i$ , $A\in nAmp(i)$, $1 ≤ i ≤ m$, $1 ≤ k ≤ f$ (the number of elements in each fold is $m = n / f$).
4. Implement a clustering algorithm (for example, k-means) on NewAmp, to return k clusters.
5. Return the mean, maximum, and minimum values of the clusters.
```

The input to the *f*-folds feature subset selection algorithm (Algorithm 1) is a set of *n* amplitudes (*Amps* = *amp*$_1$, *amp*$_2$,… , *amp*$_n$). In step 1 the algorithm divides the data set into *f* folds. All folds contain the same number of *m* amplitudes where $m = n/f$. In step 2 and 3 the algorithm forms a new set of data containing *m* records by assigning the amplitudes with the same index in all folds to the data set as one record (for example, the first amplitudes in all folds form the first record and so on). This creates the data set *NewAmp* = {*nAmp*(1), *nAmp*(2),… , *nAmp*(m)}. The number of variables in each record is *f* (the number of folds). In step 4 the algorithm implements a clustering algorithm (for example, *k*-means algorithm) on *NewAmp* to divide their instances into *k* clusters. Since each record has *f* variables, the algorithm returns *f* mean values, *f* maximum values, and *f* minimum values of each cluster. These values would be considered as representatives to the clusters and when combined together they can replace the original data set. For example, if there are 100 instances in the cluster, only 3 instances are used (means, maximums, and minimums). The total number of the variables (*t*) in each damage type would be reduced to $3\times f\times k$, when the means, maximums,

Table 1. Confusion matrix of Naïve bayes for 4 folders and 2 clusters.

Correctly classified 51 (91.0714%)					
a	**b**	**c**	**d**	**e**	**←Classifies as**
11	0	0	0	0	A
0	6	3	0	0	B
0	0	11	1	0	C
0	0	0	11	1	D
0	0	0	0	12	E

and minimums of the clusters are considered. Finally, it will be reduced to $f\times k$, if only the means are considered. The values of f and k must be determined by the user such that $t << n$, which believed to decrease the number of variables to an optimum number that highly increase the accuracy of the model and simplify it.

EXPERIMENTS SETUP

Experimental data were recorded by Worden and Lane (2001). Two data sets are used for testing as part of the methodology. The first set represents voltage amplitudes of Lamb-waves produced and collected from quasi-isotropic laminates. The second set is a vibration data from a type of ball bearing operating under different five fault conditions. The ball bearing is of the type 6204 with a steel cage. The raw measurement data took the form of an acceleration signal recorded on the outer casing for the bearing in five states.

1. New ball bearing (*a*).
2. Outer race completely broken (*b*).
3. Broken cage with one loose element (*c*).
4. Damaged cage, four loose elements (*d*).
5. No evident damage, badly worn ball bearing (*e*).

The rotational frequency was 24.5625 *Hz* and a tacho-signal was used for the measurement. The sampling frequency for the time data was 16384 *Hz* and the acquisition system was a *Bruuel* and *Kjaer* spectrum analyzer. The points were recorded in 56 instances of 2048 samples, where 11 instances for case 1, 9 for case 2, 12 for each case of 3, 4 and 5.

The pre-processing was kept to a minimum. Each signal was divided into overlapping 64 point intervals each offset by eight points from its predecessor. Each set was Fourier transformed and the magnitude of each spectral line was recorded. This yielded a sequence of 32 component vectors for classification (Jensen, 2001).

The *f*-FFE Algorithm is implemented by writing two software programs. The software used for the implementation is Java programming. To validate the results of the classifiers, it has been decided to compare the results of BN classifier to the results of NN classifiers implemented in WEKA

The first Java program will implement step 1 and 2 of the *f*-FFE algorithm. Every instance in the data set was divided by the program into different number of folders (4, 6, 8, 10 and 12). As mentioned previously, the number of samples in every instance is 2048.

The second Jave program implements step 3 of the *f*-FFE algorithm. This program was run on all files created by the first program. This program creates the mean, maximum and minimum values of the clusters after dividing them into subsets. The number of the subsets and the number of elements in each subsets are dependent on the number of folders and clusters.

EXPERIMENTAL RESULTS, ANALYSIS AND DISCUSSION

The Naïve bayes and the back-propagation *NN* classifiers found in WEKA (Witten and Frank, 2005) were implemented on the features extracted by the *f*-FFE. Labeling the instances in these files involve applying a previously learned classifier to an unlabeled data set to predict instance labels. The classifiers were firstly tested using the mean, maximum, and minimum features, secondly using the mean and maximum features, thirdly using the mean features only, and lastly using the maximum features only. The classification of the two classifiers was done for a number of clusters ranging from 2 to 8 and a number of folders, which is 4, 6, 8, and 10. The percentages of the correctly classified instances together with the confusion matrices for the classification results for each case were recorded.

Table 1 shows the confusion matrix of the classification results of the Naïve bayes for 4 folders and 2 clusters, when using the mean, maximum, and minimum features. In the table, the number of correctly classified instances is 51 out of 56 (91: 0714%).

Table 2 shows the confusion matrix of the classification results for the *BP NN* classifier for 10 folders and 2 clusters, when using all features. In the table, the number of correctly classified instances is 51 (91: 0714%). The table shows false positive and false positive classifications.

The best classification accuracies in most cases were obtained when the combination of mean and maximum features has been used. For that reason comparison between the two classifiers is limited only to the combination of mean and maximum features.

In all classifiers, the best classification accuracies were obtained when the combination of mean and maximum features and mean features only with 6 folders and 4 clusters were used. In this case the number of the features will be decreased for each instance from 2048 to 48. It has also shown that using the maximum features alone for classification will highly decrease the accuracies

Table 2. Confusion matrix of *BP NN* for 10 folders and 2 clusters.

Correctly classified 51 (91.0714%)					
a	**b**	**c**	**d**	**e**	**← Classifies as**
9	0	0	2	0	A
1	7	1	0	0	B
0	0	11	1	0	C
0	0	0	11	1	D
0	0	0	0	12	E

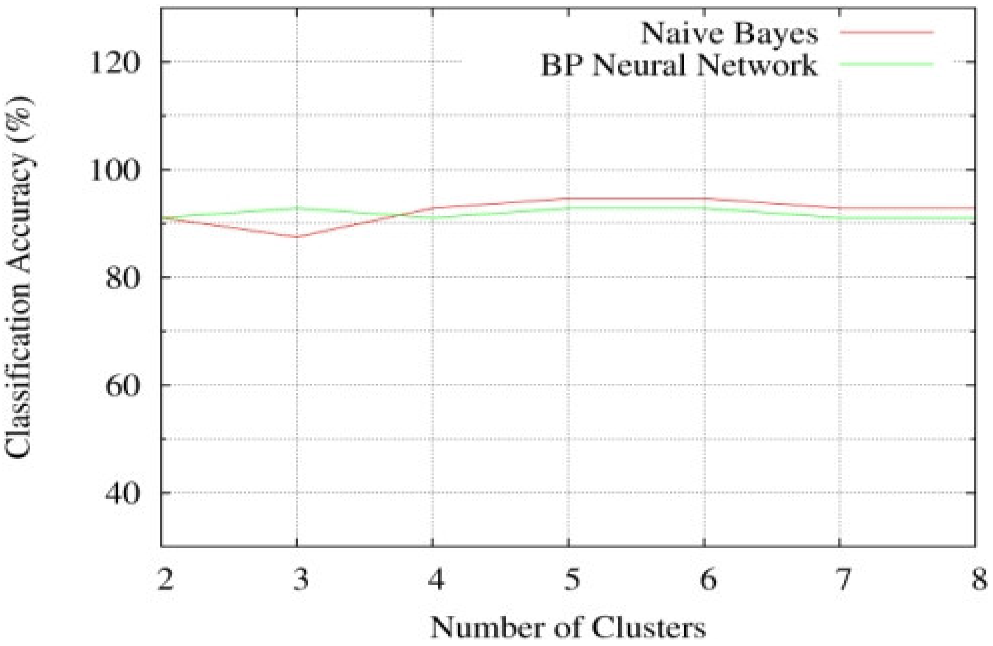


Figure 4. Classification accuracies of the classifiers when the number of folders is 4.

of the classifiers but the mean features alone have shown very good accuracies when compared to the maximum features, but less better than the combination of mean and maximum.

The *BP NN* classifier has shown many cases of false positive and false negative classifications. This is due to the nature of the data used in damage detection, where their attributes are conditionally independent given the damage attribute, which can be represented very well by Naïve bayes networks.

It has also shown that using the maximum features alone for classification is highly decreasing the accuracies of the classifiers but the mean features alone is showing very good accuracies when compared to the maximum features, but less better than the combination of mean and maximum.

Figure 4 shows the classification accuracies of the two classifiers when the number of folders is 4. It is clear the accuracy of the two classifiers is almost the same with very small variations.

Figure 5 shows the claasification accuracies, when 6 folders have been used.

The Naïve bayes classifier has not shown any case of false negative or false positive Figure 6 shows the classification accuracies of the classifiers when the number of folders is 10. The classification accuracies of the *BP NN* is a bit higher than the Naïve bayes classifier when the number of clusters is small but when the number increases the accuracies temp to be the same.

CONCLUSIONS

The research has shown the efficiency of Naive bayes and the back propagation neural network when using the *f*- folds feature extraction algorithm for damage detection ball bearings. In all classifiers, the best classification accuracies were obtained when the combination of mean and maximum features and mean features only with 6 folders and 4 clusters were used. The number of the

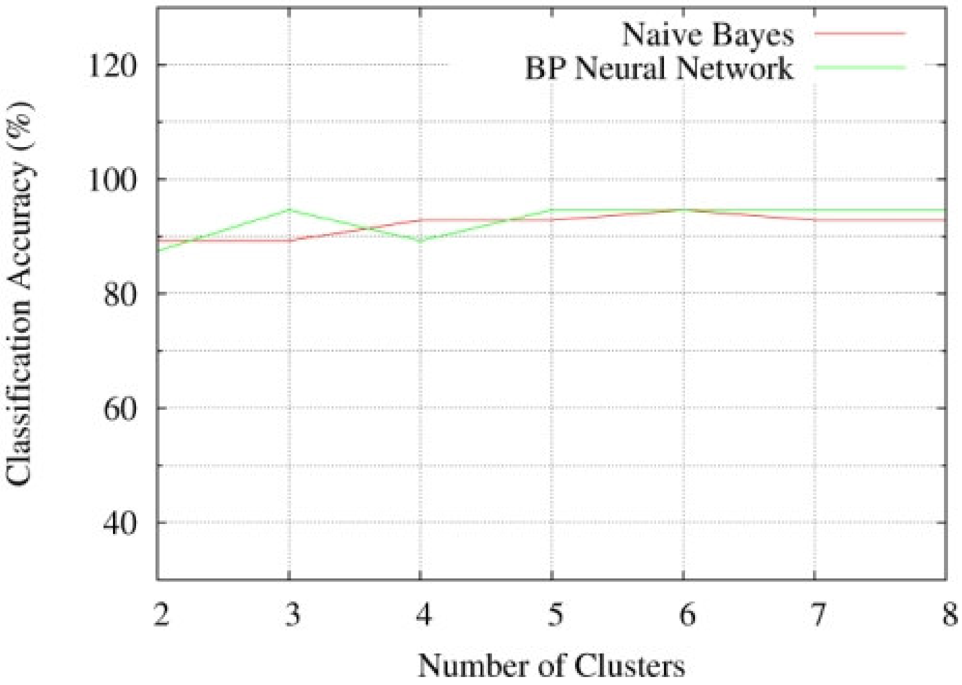


Figure 5. Classification accuracies of the classifiers when the number of folders is 6.

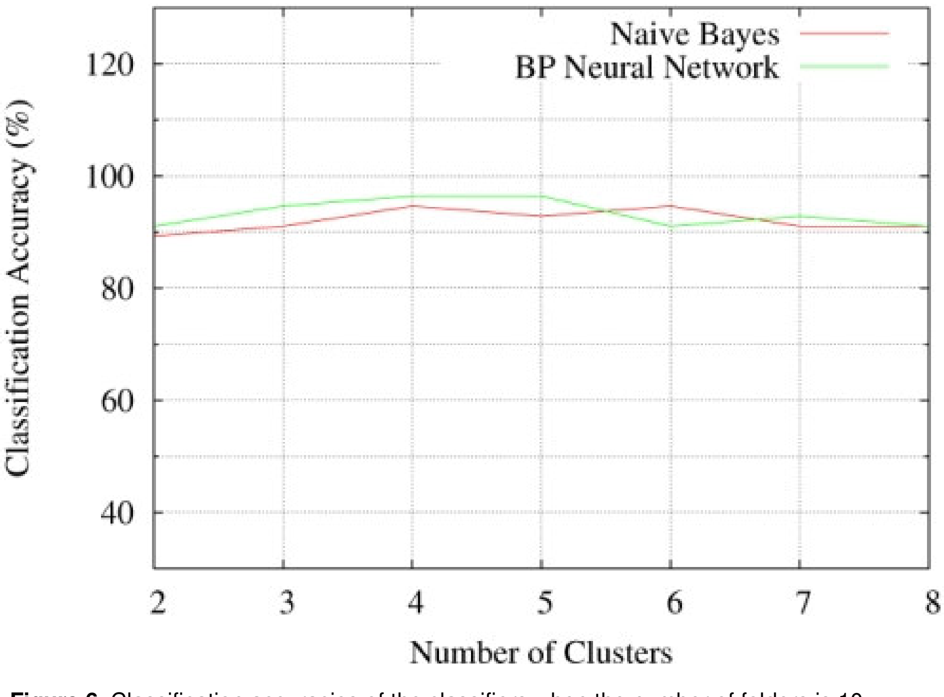


Figure 6. Classification accuracies of the classifiers when the number of folders is 10.

features was decreased from 2048 to 48. It has also shown that using the maximum features alone for classification will highly decrease the accuracies of the classifiers but the mean features alone have shown very good accuracies when compared to the maximum features, but a less better than the combination of mean and maximum. The Naive bayes classifier has not shown any case of false negative or false positive classification.

However, the back propagation neural network classifier has shown many cases of false positive and false negative classifications. This is might be due to the nature of the data used in damage detection, where their attributes are conditionally independent given the damage attribute, which match with the assumption based on the Naïve bayes classifier.

ACKNOWLEDGEMENT

The authors wish to thank Ministry of Science, Technology and Innovation Malaysia for funding the project through research grant IRPA.

REFERENCES

D'Souza K, Epureanu B (2005). Damage Detection in Nonlinear Systems using System Augmentation and Generalized Minimum Rank Perturbation Theory, Smart Materials and Structures, 14:989-1000 (2005).

Duda R, Hart P (1973). Pattern Classification and Scene Analysis, John Wiley and Sons, New York.

Guo AB, Dale WS (2005). An Experimental Investigation of White Layer on Rolling Contact Fatigue using Acoustic Emission Technique. Int. J. Fatigue, 27: 1051-1061.

Jensen FV (2001). Bayesian Networks and Decision Graphs, Springer-Verlag, New York.

Kessler S, Spearing S, Atalla M, Cesnika E, Soutisb C (2002). Structural Health Monitoring in Composite Materials using Frequency Response Methods, Composites Part B, 33: 87-95.

Witten IH, Frank E (2005). Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann, San Francisco.

Worden K, Lane A (2001). Damage Identification using Support Vector Machines, Smart Mater. Structures, 10: 540-547.