JURNAL KEJURUTERAAN
http://www.eng.ukm.my

A Novelty Detection Technique for Machine Condition Monitoring Using S.O.M.

M. L. Dennis Wong

School of Engineering
Swinburne University of Technology (Sarawak Campus)
State Complex, Jalan Simpang Tiga,
93576 Kuching, Sarawak,
Malaysia

Received Date: 28th August 2006 Accepted Date: 7th March 2007

ABSTRACT

This paper presents a novelty detection based method for machine condition monitoring (MCM) using Kohonen's self-organising map (S.O.M.). As the fault data set is difficult to acquire in MCM problems, the method requires only the knowledge of normal condition data set. By exploiting S.O.M.'s ability of multi-dimensional mapping, the Euclidean distance between the S.O.M. and the data under test is used to discriminate anomaly from normal condition. A set of real world condition monitoring data is used to evaluate the method presented. Experimental result shows high accuracy and reliability of this method.

Keywords: Novelty detection, neural network, vibration analysis, unsupervised learning, machine condition monitoring.

ABSTRAK

Kertas kerja ini membentangkan suatu pembaharuan dalam kaedah berasaskan pengesanan bagi Pemantauan Keadaan Mesin atau Machine Condition Monitoring (MCM) menggunakan Peta Penyusunan Sendiri Kohonen atau Kohonen's self-organising map (S.O.M.). Memandangkan set data kerosakan adalah sukar untuk diperoleh dalam masalah-masalah MCM, kaedah ini hanya memerlukan pengetahuan set data keadaan normal. Dengan mengeksploitasikan keupayaan S.O.M. dalam pemetaan multi-dimensi, jarak Euclidean di antara S.O.M. dan data yang diuji digunakan untuk membezakan sesuatu kejanggalan daripada keadaan normal. Satu set data pemantauan keadaan sebenar digunakan untuk menilai kaedah yang dibentangkan. Keputusan eksperimen ini menunjukkan kadar ketepatan yang tinggi dan kebolehharapan terhadap kaedah ini.

Kata kunci: Pembaharuan dalam kaedah pengesanan, rangkaian neural, analisis getaran, pembelajaran tanpa seliaan, pemantauan keadaan mesin.

INTRODUCTION

In recent years, machine condition monitoring (MCM) has undergone a transformation from a predominantly manually based monitoring approach to a highly automated approach where manual intervention is only required in the event of a fault being detected. This emerging trend has created a need for automated techniques that identify anomalies while the machine is still on-line. An early detection of possible machine failures, in some extreme cases, could save the industry millions of dollars.

Many approaches have been proposed for this purpose, from basic threshold of vibration levels (probably the most commonly used), to fuzzy logic and neural network based learning approaches [Xu et al 1997, Kim and Li. 1995, McCormick and Nandi. 1997, Murray and Penman 1997, Jack and Nandi. 2000]. Basic threshold methods require thorough investigation of the machine behaviour and characteristics. It is normally specialised and only cater specific machinery. The fuzzy logic based methods are similar to the basic threshold methods, except it offers more flexibility and offers a greater degree of intelligence. The neural networks based methods are increasingly popular as no specific background information is needed. As long as one has a labelled data set that is statistically representative, the neural network based methods normally give promising performance.

One of the main problems inherent in the current supervised neural network approach is that training data from all classes are required during the training phase of the supervised neural network. Class labels are essential in order for the decision boundaries to be drawn. However, the problem in the MCM environment is that often only sufficient information from one class may be available. The system must be able to detect when the machine condition deviates from a normal condition, and hence raise a warning. This issue is further complicated by the fact that throughout the natural life cycle of a machine the vibration characteristics will naturally change and any system used must be capable of dealing with this as part of the normal operation of the system.

In this work, a S.O.M. based novelty detector is presented for use in the MCM problem. The earliest proposal of S.O.M. in this application can be traced back to Zhang et al. 1997. With the proposed architecture, training data from only one class is used to detect deviation from that class, assuming that the training data is an adequate representation of the sample distribution. Further refinements could be added to the algorithm to allow it to operate for multiple sensors as shown in a modified version presented in Wong et al 2006.

This article is organised in the following fashion: Section 2 provides a brief introduction to S.O.M. while Section 3 discuss the application algorithm for novelty detection. Section 4 describes the dataset chosen for experiments, while Section 4 details the experiment training process of S.O.M. Finally, Section 5 provides a good indication of the capability of the proposed method through the experimental results presented. Conclusion is given in Section 6.

SELF ORGANIZING MAP

The Self-Organising Map (S.O.M.) has been used for a variety of different applications. S.O.M. is a neural network model well known for its application in high-dimensional data analysis and mapping. In the simplest sense, a S.O.M. is able to map multi-dimensional input vectors onto a simple, low dimensional grid structure, commonly of one to three dimensions. Figure 1 demonstrates this concept on a single dimension toy dataset. One might also view S.O.M. as a non-linear projection of the probability density function of the higher dimensional input data onto a low dimensional display [Flexer 1999, Vesanto 1999].

The map consists of a set regularly shaped map units or neurons. The topology (the way that the neurons are arranged) of the neurons is commonly rectangular or hexagonal (see Figure 1). Depending on the application, it can also be a random grid as in the example used in this paper. Each neuron has a neighbourhood function that specifies its relationship to its neighbouring neurons. This neighbourhood function can be a Gaussian function among others:

$$h_{j,i}(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right) \tag{1}$$

where $h_{j,i}$ denotes the influence of neuron j by the winning neuron i ; i denotes the effective width of $h_{j,i}$ which varies according to the time index n.

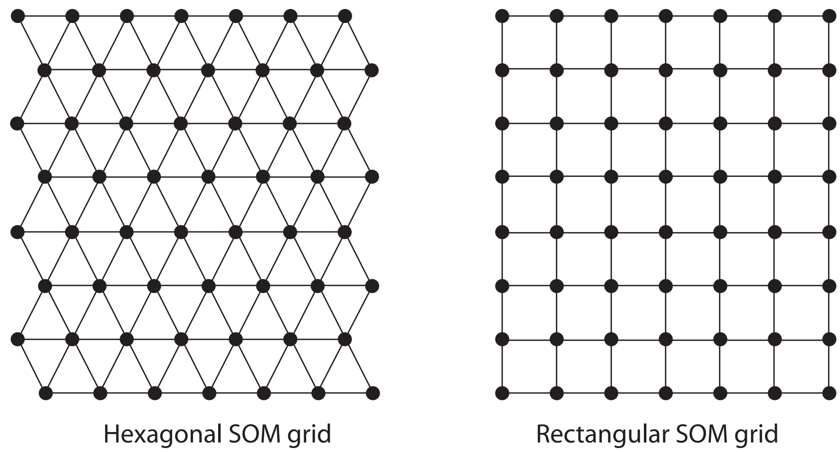

FIGURE 1. SOM can exists as regular hexagonal or rectangular grid

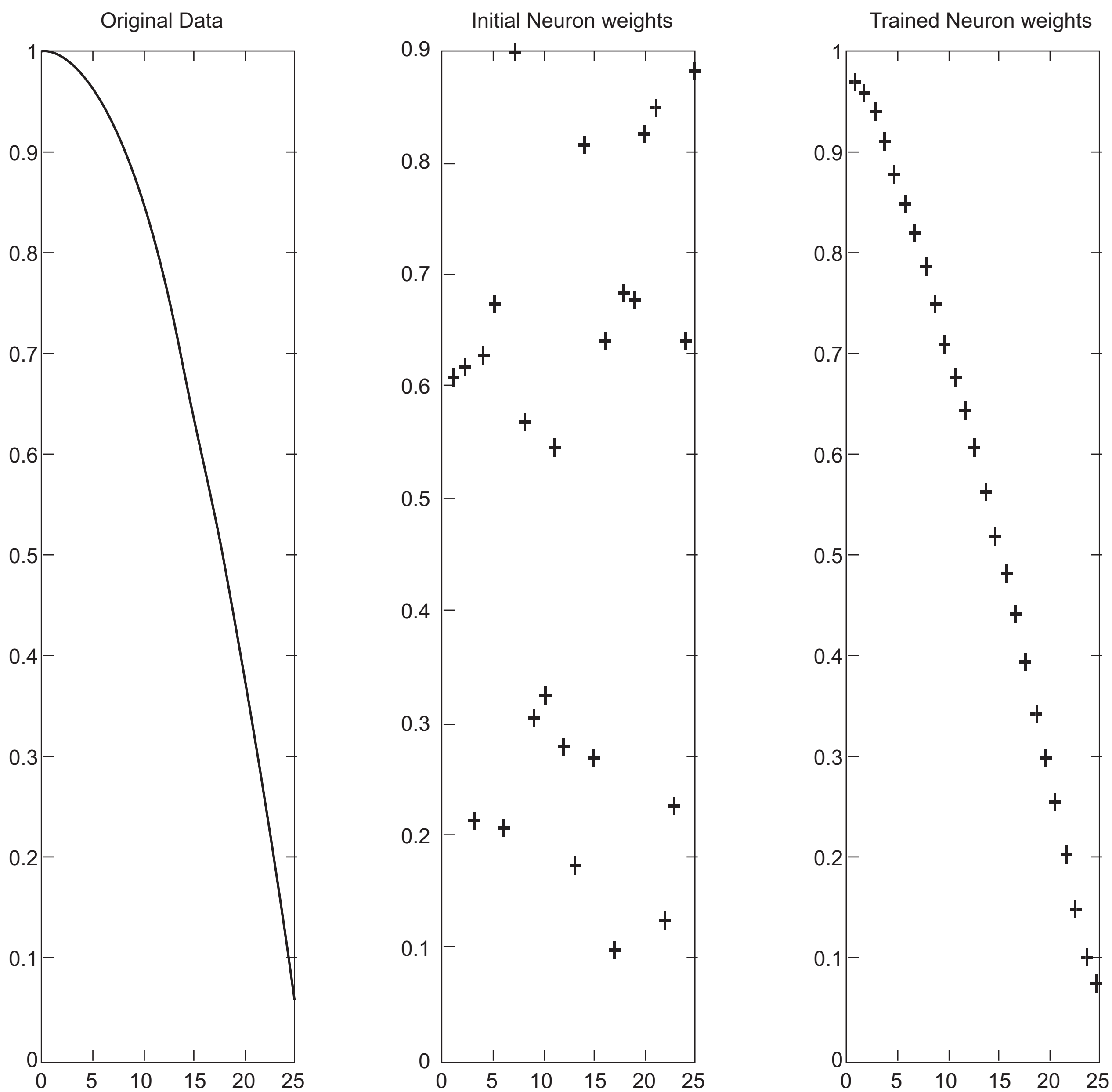

FIGURE 2. Mapping of S.O.M. in one dimension. Left figure shows the data to be mapped. The middle figure shows the initial neuron positions, while the right figure shows the neurons' positions after training

The training of the S.O.M. is done by minimising a metric between its neurons and the input vectors. One practical example of such a metric is smallest distance measure and its neighbouring neurons (defined by $h_{j,i}(n)$) are moved towards the input vector:

$$w_j(n+1) = w_j(n) + \eta(n)h_{j,i}(x - w_j(n)) \quad (2)$$

where w_j is the weight of neuron j; and η is the learning rate.

Over time, this causes the network of neurons to effect a mapping of the training data onto a one or two-dimensional structure. More detailed description of the S.O.M. training algorithm can be found in [Kohonen 1997, Haykin 1999]. An example of a one dimensional S.O.M. mapping onto a single curve is shown in Figure 2.

S.O.M. FOR NOVELTY DETECTION

In this section, the demonstration of S.O.M. for novelty detection is presented. In order to map the distribution of the normal data, a set of sufficiently large collection of training data (containing only the normal condition feature vectors) is presented to the S.O.M. The method for a single sensor environment is presented and depicted in Figure 3. Essentially, there are three steps in the training of S.O.M., i.e. i) data normalisation, ii) training of S.O.M. and iii) Calculation of threshold. For testing, two other steps aperformed, i) presentation of test set and ii) classification base on the quantization error.

Prior to the training process, a suitable method of normalization must be performed for the training data. There exists a variety of normalization methods, e.g. histogram normalization, variance normalization, range normalization etc. One common method is to normalize the data so that it is of zero mean and unit variance. This could be achieved by performing the following:

$$\hat{x} = \frac{x - \bar{x}}{std(x)} \quad (3)$$

where $\bar{x}$ refers to the mean of x and std(•) denotes the standard deviation function.

This normalization process is essential to ensure the range of the training data is suitable for the training process. At the same time, it increases the speed of the training process and helps to minimize the topographic error between the map and the data set.

After the normalization, the data is used to train the S.O.M. The length of the training process depends on the complexity of the given problem. The primary factor of the training process is the dimensionality (degree of freedom) of the data. The higher the dimension, the longer the training process is. At the end of each training epoch (a complete cycle through the available data vectors), two quality control variables are calculated, i.e. the quantization error and the topological error. The stopping criterion for the training process could then be based on either one of the two variables. Alternatively, the stopping criterion could be a fixed number of epochs.

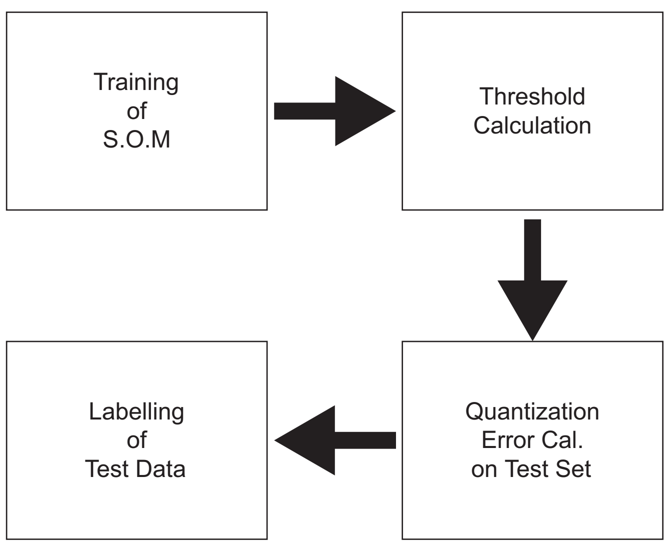

FIGURE 3. S.O.M. for novelty detection

In this paper, the S.O.M. is trained for a fixed number of epochs, and the quantization error is used to assess the quality of the map. After the mapping, the quantization errors of each training example with the trained S.O.M. are calculated and its statistical mean and standard deviation is then recorded. One can then calculate the threshold, T, as follows:

$$T = (1+\mathrm{std}(E_q))\bar{E}_q \quad (4)$$

where denotes the quantization error vector. The calculated T can then serve as a control parameter to determine if a given data set belongs to the normal data. The S.O.M. novelty detector is now ready to be deployed.

To test the accuracy of the detector, a set of testing data set consisting of both normal and abnormal data vectors is presented to S.O.M. One can then calculate the quantization error between the map and the data vector under test. If the quantization error calculated exceeds the previously calculated threshold, T, the data vector under test is then labelled as an anomaly. Otherwise it is labelled as normal.

EXTENSION TO MULTI-SENSOR ENVIRONMENT

The above proposed Novelty detector could be easily extended to a multi-sensor environment. Many times the machine under test has more than a single sensor. Generally, the feature vectors extracted from the original time series are multi-dimensional (as illustrated in Section 4 later). One simple way is to cascade the feature vectors from different sensors together and form a new feature vector. This how ever increase the number of dimension of the feature vector which could possibly slow down the training process and could, at worse case, affect the convergence of the algorithm.

A better way of dealing with multi-sensor environment is to deploy a S.O.M. novelty detector at each sensor, and have another metric to govern the output the individual detectors. By choosing an appropriate metric, one could use theoretical threshold instead of an empirical threshold. More information and thoughts on this issue could be found in at the author's other works [Wong et al 2006].

EXPERIMENTAL SETUP AND DATASET

To illustrate the usage of the presented method for novelty detection, a public domain data set for condition monitoring was chosen. The dataset can be downloaded from the Structural Integrity and Damage Assessment Network (http://www.sidanet.org). It covers detection of four types of fault in ball-bearing cages as illustrated in Error! Reference source not found.. Ball bearing data was chosen because it is a critical component in axles in vehicles, machines and safety-critical systems such as aircraft wing flaps.

Each example originally consisted of 2048 samples of acceleration taken with a Bruel and Kjaer vibration analyser. A FFT operation was then performed on each example to give a 32-point FFT as input features to the S.O.M.. The normal dataset for training consists of 913 examples while for the test dataset there are 913 examples for normal condition, 747 examples for Type I fault, 996 examples each for Type II, III, and IV faults.

The normal dataset was used to train the S.O.M. using different parameters of the S.O.M.. These parameters include the topologies of the S.O.M., shapes of the S.O.M., the training and fine tuning epochs of the S.O.M. After training, the test dataset was used to test the performance of the proposed algorithm. The simulation was conducted in MATLAB environment with the aid of SOMTOOLBOX V2.0 (http://www.cis.hut.fi/projects/somtoolbox/).

TABLE 1. Types of faults and their respective amount available in the test set constructed

Type of Fault	Description	No. of Examples in Test Set
I	outer race completely broken	747
II	fault: broken cage with loose element	996
III	damaged cage with 4 loose elements	996
IV	badly-worn ball bearing with no evident damage	996

^ ‾

RESULTS AND DISCUSSION

Figure 4 shows a simple visualization aid to view the trained S.O.M. using only the first two dimension of the input feature vector. The diagonal shows to distances between each neuron in the Map in that dimension, while the top left figure shows the reference colour map. The 2nd and 3rd column figures in the first row shows the histogram of the neurons, while the 2nd and 3rd row figures in the first column shows the histogram of the training data. The two scatter plots show the 2D distribution of the training data and the S.O.M. itself respectively. It can be concluded from the diagram the S.O.M. perform a good mapping while maintain adequate generalization.

Figure 5 depicts the output (quantization errors) of the S.O.M. during a typical round of test process. The first 913 samples (Normal condition) is clearly lower than the rest of the rest of the samples (Abnormal Condition). This demonstrated the intrinsic idea of adopting S.O.M. to reduce the 32 dimension into 1 single metric for classification.

Table 2 shows the performance of the S.O.M. using a HEXTOP topology and trained and fine-tuned for 100 epochs each. The figures shown are the average value after 100 independent run with a standard deviation of 0.17% and 0.13% on Normal and Abnormal condition respectively. It is shown that the algorithm can detect the fault condition consistently and accurately.

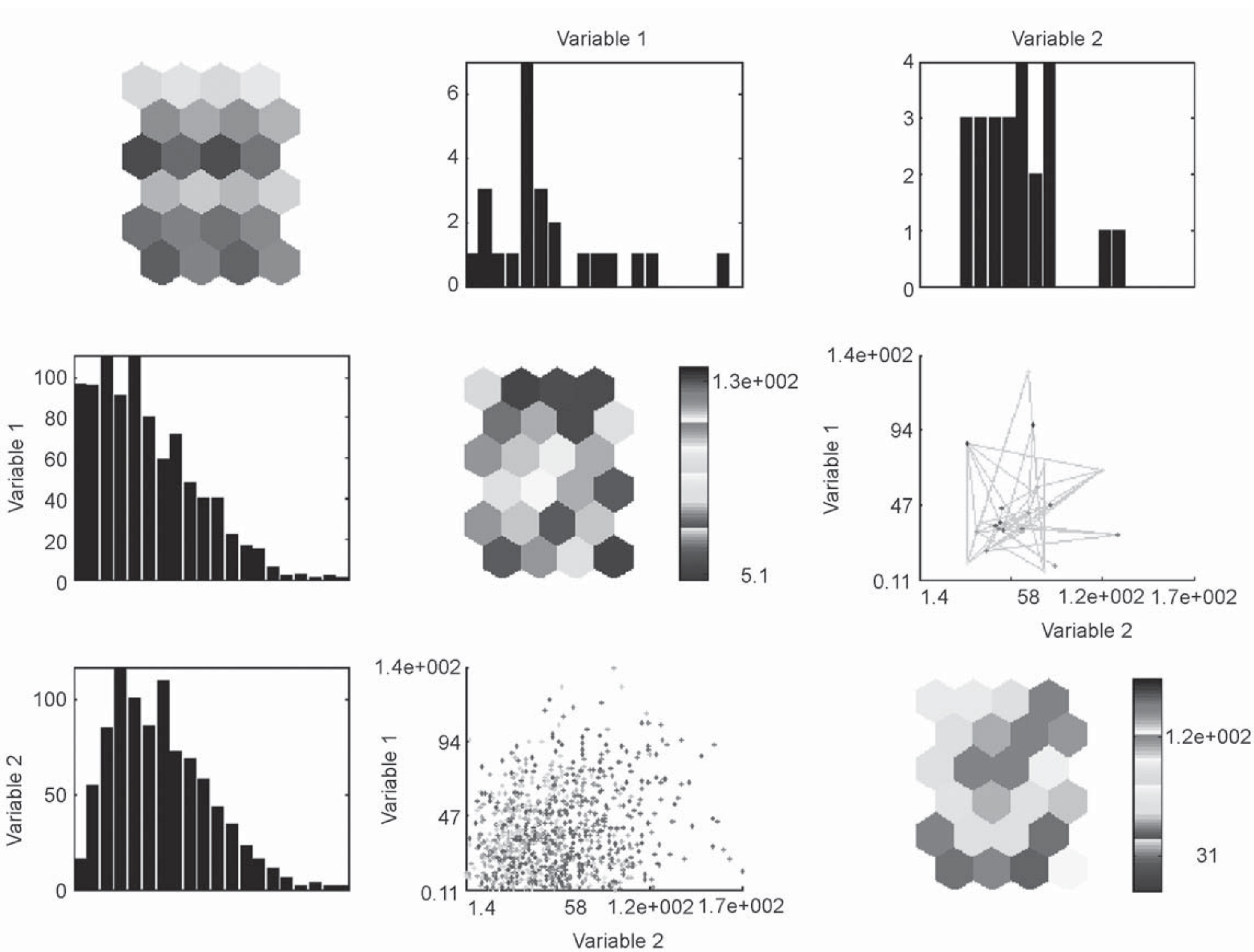

FIGURE 4. A Visualization of the trained S.O.M. using only the first 2 components

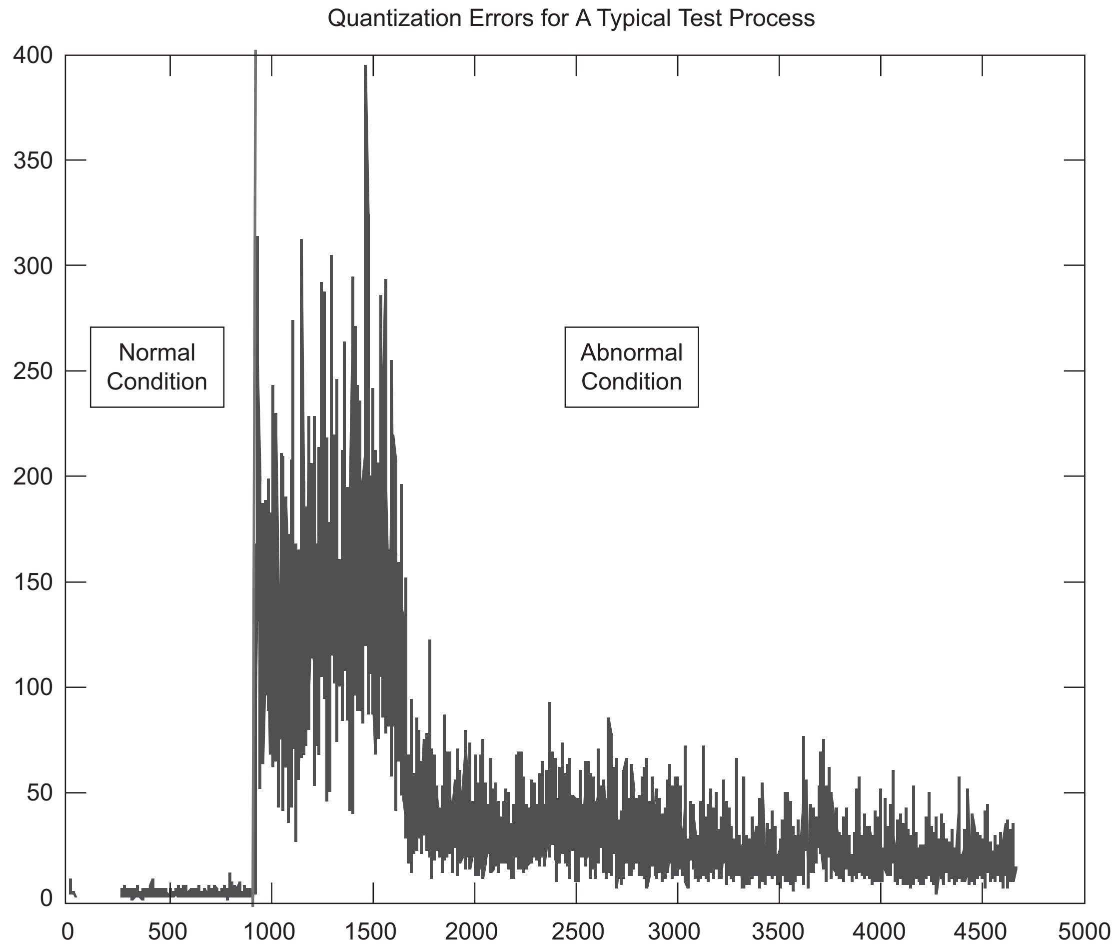

FIGURE 5. Output quantization of S.O.M. during the testing process. The red dashed line depicts the separation on normal and abormal samples. The X-axis denotes the sample no while the Y-axis is the qerror metric

TABLE 2. The confusion matrix of the proposed algorithm (in %). Table shows average result obtained after 100 rounds of simulations

Perceived Condition	Actual Condition	
	Normal	Abnormal
Normal	99.7	0.8
Abnormal	0.3	99.2

CONCLUSION

Novelty detection has many applications in industries. Here, the author presented a recent method in MCM based on the concept of novelty detection using S.O.M. The proposed method for single sensors is presented but it could be easily extended to multi-sensors environment. The modularity of the method enables it to be adapted in other cross-disciplinary applications. The simulation results shown the method possesses good accuracy and it is robust against different parameter settings of the S.O.M. Evaluation using a real world dataset shows 99% successful classification.

REFERENCES

A. Flexer 1999. On the use of self-organizing maps for clustering and visualization, in *Principles of Data Mining and Knowledge Discovery*, pp. 80-88.

S. Haykin 1999, *Neural Networks*, 2nd Edition, Prentice Hall, Chapter 9.

L. B. Jack and A. K. Nandi. 2000. Comparison of neural networks and support vector machines in condition monitoring applications, in: H. C. Pusey, R. B. K. N. Rao (Eds.), *Proceedings of COMADEM* 2000, MFPT Society, Willowbrook, Illinois, USA, pp. 721-730.

T. Kim and C. J. Li. 1995. Feedforward neural networks for fault diagnosis and severity assessment of a screw compressor, *Mechanical Systems and Signal Processing* 9 (5) pp. 485 496.

T. Kohonen, 1997. Self-Organizing Maps, Springer-Verlag.

A. C. McCormick and A. K. Nandi. 1997. Neural network autoregressive modelling of vibrations for condition monitoring of rotating shafts, in: *Proc of the 1997 Int. Conf. on Neural Networks*, Houston, Texas. pp. 2214-2218.

A. Murray and J. Penman 1997. Extracting useful higher order features for condition monitoring using artificial neural networks, *IEEE Trans. on Sig. Proc.* 45 pp. 2821-2828.

M. L. D. Wong, L. B. Jack, and A. K. Nandi. 2006. Modified self organising map for automated novelty detection applied to vibration signal monitoring. *Mechanical System Signal Processing* ISSN: 0888-3270, 20(3): 593--610,

J. Vesanto 1999. S.O.M.-based data visualization methods, Intelligent Data-Analysis 3, (1999) pp. 111-126. URL:http://citeseer.nj.nec.com/vesanto99sombased.html

X. Xu, H. Vanderveldt, and R. Allen. 1997. An ANS based helicopter transmission diagnostic system, in: *Proc. of the 1997 Int. Conf. on Neural Networks*, Houston, Texas, pp. 2214-2218.

S. Zhang, R. Ganesan, and G. D. Xistris. 1996. Self-organising neural networks for automated machinery monitoring systems, *Mechanical Systems and Signal Processing* 10 (5) pp. 517 532.