

Visualization of Induction Machine Fault Detection Using Self-Organizing Map and Support Vector Machine

Sitao Wu, Tommy W. S. Chow, Di Huang

Department of Electronic Engineering, City University of Hong Kong, Hong Kong

Abstract: Induction machines play an important role in today's industries. How to monitoring, detection, classification, and diagnosis of induction machine faults have been the essential problems. Although there have been many methods proposed to deal with these problems, there is lack of visualization tool for understanding the problems more easily. In this paper, a visualization method is proposed to help users understand the mechanism of induction machine fault detection in a transparent way. Furthermore, user can also tell the status (normal or faulty) just directly from the visualization results. The visualization is implemented by hybridizing two neural networks: self-organizing map and support vector machine. Experimental results demonstrate the novelty and effectiveness of the proposed visualization method used for induction machine fault detection.

Keywords: Self-organizing map (SOM), support vector machines (SVMs), SVM visualization (SVMV), visualization

I. INTRODUCTION

On-line machine fault detection is widely used for providing safe and economical industrial operation [1-3]. Among the many different types of faults diagnostic methods [4], the use of vibration signals has been recognized as a reliable approach as they are based on easily accessible and non-destructive measurements. Recently artificial intelligence (AI) techniques have been proposed for the noninvasive machine fault detection [5-6]. These AI based techniques include expert systems, and neural network (NN). Expert system uses knowledge-based rules to model the system [6]. NN approaches can be considered as "black-box" methods as they do not provide reasoning about the fault detection process [5][7]. In the NN type approach, BP and RBF supervised NN have been used in induction machine fault detection [8-12]. In the field of NN classification, support vector machine (SVM) [13] is one of the widely used classification methodologies recently because its superior performance over other methods like BP and RBF NN.

Furthermore, the above NN type approaches are difficult for non-NN experts to explain why they have such a good capability of classification of input data with dimension more than three. If the NN type approach is used with the interpretation by a visualization method, the mechanism of "black box" in NNs can be reduced to some extent. The objective of visualization is to project high-dimensional data to low dimensions (usually 2-D). Popular choices for visualization are classical dimensionality reduction techniques such as principal component analysis (PCA) [14], multi-dimensional scaling (MDS) [15], self-organizing mapping (SOM) [16], etc. PCA is a linear dimension reduction technique and is deemed unsuitable for handling very complicated data. Some nonlinear PCA [17] and kernel PCA [18] were proposed to address this problem. MDS, a family of nonlinear dimension reduction techniques, produces geometric representation of data in low dimensions. Sammon's mapping [19] is one of the most widely used MDS methods. Curvilinear component analysis (CCA) [20], ISOMAP method [21] and LLE method [22], extend the traditional MDS methods by considering local similarity measure. SOM is an unsupervised and non-parametric NN method that can be used for clustering and visualization. All the above mentioned visualization methods do not utilize class information during the course of computation, i.e., they are unsupervised methods. When a dataset with class information, class labels can be displayed in the reduced low-dimensional space as a post-process after the completion of training. But the boundary between classes of data can hardly be determined in the reduced low dimensional space.

SVM provides good classification performance but cannot provide visualization on its results, whereas SOM is an unsupervised NN that provide visualization with little boundary information between classes. In this paper, a new visualization method, called SVM visualization (SVMV), is proposed to visualize the SVM classification result on induction machine fault detection. The SVMV method is developed based on the advantages of both SVM and SOM. Unlike the abovementioned visualization methods, SVMV provides direct boundary between classes, which is important in understanding machine faults classification. Compared with the support vector visualization and clustering (SVVC) algorithm [23] that provides only the visualization of the boundaries of all input data without using class information, the proposed SVMV algorithm gives more information about classification boundary. That is, the faults and normal conditions lying close to the class boundary, which are interesting to us, can be easily classified just from visualization. Furthermore,

with the help of the proposed SVMV algorithm, the distance between a datum and classification boundary can be directly visualized in a low-dimensional space. Despite the fact that there is always a possibility of mis-classification for the data close to the classification boundary, with this newly proposed methodology we are now able to directly classify data simply according to the low-dimensional visualization, i.e., visual classification. Consequently, unlike SVM that is performed as a black box and difficult to explain the mechanism of classification, the classification results can be clearly elaborated through visual classification of SVMV. In the next section, the proposed SVMV method is described in detail based on the introduction of SOM and SVM. In section III, experimental results on a real induction machine by the proposed SVMV algorithm will be presented. Finally, conclusion is drawn in section IV.

II. The SVMV method

A. Introduction of SVM

SVMs are based on statistical learning theory [13]. SVMs have been successfully applied in a variety of classification field. There are mainly two SVMs models for binary classification: C-SVMs [13] and ν -SVMs [24]. ν -SVMs have similar performance with C-SVMs. However, the parameter ν in ν -SVMs indicates the lower bound of the ratio between the number of support vectors and that of the whole input data, and the upper bound of classification error in training data [24], while the parameter $C \in [0, \infty)$ in C-SVMs is not convenient to select. Therefore ν -SVMs are better than C-SVMs in that the parameter $\nu \in (0, 1)$ in ν -SVMs has salient physical meaning and is adopted in this paper. The following is a simple introduction of ν -SVMs. Suppose an empirical dataset $(x_1, y_1), \dots, (x_n, y_n)$, where $x_i \in \mathfrak{R}^d$ is a d -dimensional input vector, $y_i \in \{1, -1\}$ is the binary class label, n is the number of input data. ν -SVM tries to separate two classes by using an optimal hyperplane that maximizes the margin of separation. It solves the following primal problem:

$$\begin{cases} \min & \phi(W) = \frac{1}{2} \|W\|^2 - \nu\rho + \frac{1}{n} \left(\sum_{i=1}^n \xi_i \right) \\ \text{s.t.} & y_i[(W \cdot \varphi(x_i)) + b_0] \geq \rho - \xi_i, \nu > 0, \xi_i \geq 0, \rho \geq 0, i = 1, 2, \dots, n \end{cases}, \quad (1)$$

where φ is a mapping function from input space to high-dimensional feature space, $W \in \mathfrak{R}^d$ is a vector perpendicular to the optimal hyperplane in feature space, $b_0 \in \mathfrak{R}^d$ is a bias vector, ξ_i is a slack variable for allowing classification errors, $\rho \in \mathfrak{R}^d$ is another bias vector, and ν is a parameter discussed before.

Instead of solving problem (1) directly, ν -SVMs solve the dual problem as follows:

$$\begin{cases} \max & Q(\alpha) = -\frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \varphi(x_i)^T \varphi(x_j) \\ \text{s.t.} & 0 \leq \alpha_i \leq \frac{1}{n}, \sum_{i=1}^n \alpha_i y_i = 0, \sum_{i=1}^n \alpha_i \geq \nu, i = 1, 2, \dots, n \end{cases} \quad (2)$$

where α_i is a Lagrange coefficient. The solution of α_i in (2) can be used to compute w , b_0 , and ρ in (1) [24]. To avoid computing dot product in the high dimension feature space, SVMs use kernel functions as a trick [13]. After the completion of optimization (2), the data points with $0 < \alpha_i < 1/n$ are called support vectors (SVs).

The decision function for the binary classification of the data is $\text{sign}(W \cdot \varphi(x) + b_0) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i K(x_i, x) + b_0\right)$ (3)

The bias function for a new input z is $f(z) = \left(\sum_{i=1}^n \alpha_i y_i K(x_i, z) + b_0 \right)$. The distance between z and the optimal hyperplane (classification boundary) in feature space is $|f(z)|/\|w\|$. For the SVs with $0 < \alpha_i < 1/n$, the value of $f(z)$ is ρ or $-\rho$, and that of $|f(z)|/\|w\|$ is $\rho/\|w\|$. For the data points lying on the classification boundary, the value of $f(z)$ or $|f(z)|/\|w\|$ should be zero.

B. Introduction of SOM

SOM consists of m neurons located on a regular low dimensional grid, usually a 2-dimensional rectangular or hexagonal grid. Higher dimensional grids are possible, but they are not generally used since their visualization is problematic. The basic SOM algorithm is iterative. Each neuron i has a d -dimensional weight vector $w_i = [w_{i1}, \dots, w_{id}]$ in input space. At each training step t , a sample vector $x(t)$ is randomly chosen from training sets. Distances between $x(t)$ and all weight

vectors w_i 's are computed. The winning neuron, c , has the weight vector w_c closest to $x(t)$ in the Euclidean distance, $c = \arg \min_i \|x(t) - w_i\|$, $i \in \{1, \dots, m\}$. (4)

A set of neighboring neurons of the winning neuron c is denoted as N_c , which decreases its neighboring radius $r(t)$ of the winning neuron with time. $h_{ic}(t)$ is defined as the neighborhood kernel function around the winning neuron c at time t . It is a non-increasing function of time and of the distance between neuron i and the winning neuron c . Usually the kernel

can be chosen as a Gaussian function, $h_{ic}(t) = \exp\left(-\frac{\|Pos_i - Pos_c\|^2}{2r(t)^2}\right)$, (5)

where Pos_i and Pos_c are the coordinates of the neurons in the fixed grid, respectively. The weight update rule in the

sequential SOM algorithm can be written as, $w_i(t+1) = \begin{cases} w_i(t) + \varepsilon(t)h_{ic}(t)(x(t) - w_i(t)), & \forall i \in N_c \\ w_i(t), & \text{otherwise} \end{cases}$, (6)

where $h_{ic}(t)$ is the neighborhood function around the winning neuron c and $\varepsilon(t)$ is the learning rate at time t . During the course of training, SOM behaves like a flexible net that folds onto the "cloud" formed by training data. Due to the neighborhood relations, neighboring neurons are pulled to the same direction making the weight vectors of neighboring neurons resemble each other.

The neurons of SOM in the low dimensional output space represent visualization in the input space. To visualize the original input data, one usually assigns all input data to their corresponding nearest neurons in input space and then uses neurons for visualization. Thus some input data may be superimposed in the same neuron for visualization. SOM can be improved for better visualization purposed ViSOM [25] is a new algorithm to preserve topology as well as inter-neuron distances. The final map can be seen as a smooth net embedded in input space. The distances between any pairs of neurons in input space resemble those in output space. ViSOM uses the same network architecture as SOM. The only difference between SOM and ViSOM is that the neighboring neurons of winner neuron are updated differently. In SOM, the weight-updating rule is (6). The weight-updating rule for the neighboring neurons of winner neuron in ViSOM is

$w_i(t+1) = w_i(t) + \varepsilon(t)h_{ic}(t)\left([x(t) - w_c(t)] + [w_c(t) - w_i(t)]\left(\frac{d_{ci} - \lambda\Delta_{ci}}{\lambda\Delta_{ci}}\right)\right)$, $\forall i \in N_c$ (7)

where d_{ci} and Δ_{ci} are the distances between the neuron c and i in input space and output space, respectively, and λ is a resolution parameter.

C. The proposed SVMV method

Before the introduction of SVMV method, we give the concept of interpolation of SOM used later. The computational complexity of per iteration of the whole training data for SOM is $O(mn)$, where m is the number of output neurons and n is that of input data. To obtain a more precise map and reduce the computational complexity, we can interpolate data between the neighboring pairs of weights $\{w_i\}$ in SOM and therefore extended weight vectors $\{w'_i\}$ are formed [23]. Since SOM preserves topology of input data, the interpolation is feasible. In this paper, we use rectangular grids. Only neighboring weights are considered for interpolation. The interpolation is illustrated in Fig.1. First compute the middle points between any neighboring pair of weights. For example, the interpolated weights e, f, g, h are computed by

$$w_e = (w_a + w_d)/2, w_g = (w_c + w_d)/2,$$

$$w_f = (w_a + w_b)/2, w_h = (w_b + w_c)/2.$$

Then compute the interpolation points at the centers of four neurons (a, b, c and d), which lie on the four corners of the rectangle on the 2-D grid. For instance, the interpolated weight w_i is computed by $w_i = (w_e + w_f + w_g + w_h)/4$. If the old weight matrix is $L \times L$, the new extended SOM grid becomes $(2L-1) \times (2L-1)$. If we continue to recursively interpolate the obtained SOM grid, the extended SOM grid will become $(4L-3) \times (4L-3)$, $(8L-7) \times (8L-7)$, etc.

Then the detailed procedure of the SVMV algorithm is described as follows:

Step 1) First, an SOM algorithm is trained to obtain the topology-preserved weights $\{w_i\}$, $i=1, \dots, m$.

Step 2) Then ν -SVM is used to obtain the optimal value of α_i , $i=1, \dots, n$, according to (2).

Step 3) Interpolation is performed between the neighboring pairs of weights $\{w_i\}$ in SOM and therefore extended weight vectors $\{w_i'\}$ are formed to obtain a more precise map.

Step 4) After that, the bias function $f(w_i')$ on the extended weight vectors $\{w_i'\}$ is computed according to

$$f(w_i') = \sum_{i=1}^n \alpha_i y_i K(x_i, w_i') + b_0. \quad (8)$$

Step 5) Finally a 2-D map of SOM is colored according to the values of $\{f(w_i')\}$. The neurons with grey color, whose bias $f(w_i')$ is larger than 0, belong to class "1". On the contrary, the neurons with white color, whose bias is less than 0, belong to class "-1". Obviously, the bias $f(w_i')$ of the neurons on the boundary is equal to zero. So the classification boundary can be detected by $f(w_i') = 0$. Due to the topology preservation of SOM, if a datum in the input space is close to classification boundary, it is also close to that in the reduced low-dimensional space. So the distance between data and classification boundary can be displayed on the reduced low-dimensional space.

If the SOM NN in the SVMV algorithm is replaced by ViSOM, this is a ViSOM-based SVMV.



Fig. 1. Interpolation of SOM.

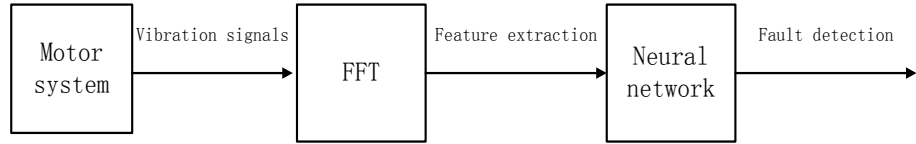


Fig. 2. Flow chart of fault detection system in frequency domain

III. EXPERIMENTAL RESULTS

A. Input features of classification

Vibration analysis is based on the principle that faults can be detected in characteristic frequencies associated with particular type of faults in frequency domain. The detection system in frequency-domain is briefly shown in Fig. 2. First vibration signals are collected from transducer-based data acquisition systems. The signals are transformed by fast Fourier transformation (FFT) into signals in frequency domain, which can be analyzed and processed easier than those in the time-domain. Then some features are extracted from the frequency-domain signals as inputs of NNs. Through supervised training with inputs and outputs, the learned NNs can detect different types of faults.

We need to reduce frequency components into fewer useful features. One way is to extract features at characteristic frequencies associated with certain type of faults [11]. Another way compresses the spectrum while nearly maintaining the shape of the spectrum [9]. The first one requires knowledge of the characteristic frequencies in the type of machine faults involved but in practical situations, many unexpected conditions make theoretic analysis difficult and inconsistent. The second one uses the shape of the spectra of the signals but the compression of the spectrum [9] is complicated and time-consuming. We use four features extracted from the power spectra of the vibration signals [12]. When the machines are in faults, the shapes of the frequency distribution of energy drift from the normal condition. The spread (dispersion) of distribution about central value (central moment) gives information about the shape or dispersion of the power spectra about its average frequency. The dispersion indices include normalized second and third order central moments [26]. By extracting the total power, average frequency, and dispersion indices of the power spectra of the vibration signals, we can detect different types of faults. The definitions of them [12] are total power: $TP = \int_0^{\infty} P(f)df$; average frequency:

$\bar{f} = \int_0^{\infty} fP(f)/TP$; normalized second order central moment: $NU2 = \int_0^{\infty} (f - \bar{f})^2 P(f)df / TP$; normalized third order central moment: $NU3 = \int_0^{\infty} (f - \bar{f})^3 P(f)df / TP$. The complete input vector is $X = [TP, \bar{f}, NU2, NU3]$.

B. Electrical and Mechanical faults

Unbalance fault of a three-phase machine is used to demonstrate the effectiveness of the proposed method. To simulate the electrical unbalance faults in a three-phase induction motor, a noninvasive test rig shown in Fig. 3 is used. As faults associated with the unbalanced phase alter the electromagnetic flux, the electromagnetic force acting upon the stator core changes. The detection of unbalance faults is based on the stator core vibration signals. The rated power and speed of the induction machine are 1100W and 1440 rpm. The case of running at 50 Hz supply frequency is studied. To simulate

unbalanced electrical faults, the “C” phase is serially connected through a variable resistor ranging from 0Ω to 50Ω to introduce unbalanced electrical faults. The machine is driven under different asymmetrical faults by different values of the resistor, i.e., 10Ω , 20Ω , 30Ω , 40Ω and 50Ω . The vibration signal is measured by an accelerometer mounted on the machine. The sensitivity and upper frequency limit of the accelerometer is $1\text{ pC}/\text{ms}^2$ and 12k Hz. A charge amplifier with the gain of 316 mv/unit amplifies the signal picked up by the accelerometer. The acquired data are fed into a low pass filter with bandwidth of 3kHz. The amplified and filtered signal is sampled and input into a Pentium PC. The sampling rate is 10k Hz with 1024 samples for each data frame, which are used as feature vectors as the NN inputs.

The major faults in the mechanical faults can be broadly classified as follows [4][9]: Stator faults, rotor imbalance, air gap irregularities, and bearing failures. We simulated the mechanical fault by adjusting the mechanical looseness through a screw used for fixing the machine. When the screw is firmly fixed and the three phase currents are symmetrical, the machine is in the normal condition. Five different levels of looseness are considered from light to heavy. The vibration signals should change to some extent according to the degree of looseness.

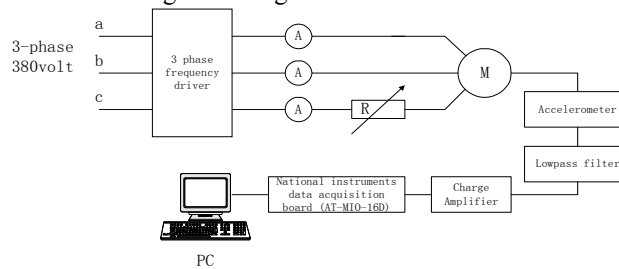


Fig.3. Testing rig of 3-phase induction motor.

C. Experiments by SVMV

The training data consist of 100 data with normal conditions, 100 data with electrical faults and 100 data with mechanical faults. The testing data consist of 50 data with normal conditions, 50 data with electrical faults and 50 data with mechanical faults. The number of input dimensions is four. The data with normal samples are labeled with class “-1” while the data with faults are labeled with class “1”. OSU_SVM (<http://svm.sourceforge.net/>) was used for the implementation of ν -SVMs. Radial basis function (RBF) kernel, the most common type of kernels used in SVM, is used here and expressed as $K(x, y) = \exp(-\|x - y\|^2 / (2\sigma^2))$. Classification performance is strongly affected by the width of RBF kernel. If σ is too large, classification hyperplane is over-smoothed and some useful boundary information is lost. If σ is too small, training may be over-fitted, lots of support vectors are obtained and the generalization ability decreases. Classification performance is also affected by the value of ν , which is a parameter controlling the fraction of support vectors and training error [24]. In this study, the typical value of ν and σ were selected as 0.05 and 50 respectively. A 20×20 SOM grid is used in the experiments. We used 2000 training iterations to train the SOM for the self-organizing phase and 2000 iterations for the convergence phase. The learning rate gradually decreases from 1 to 0.001.

The visualization of training and testing data by SOM-based SVMV is shown in Fig. 4 (a) and (b), where the normal condition, electrical faults and mechanical faults are visualized by black, red and green symbols, respectively. The SVM classification accuracy for the training data is 97.0% and that for the testing data is 96.0%. The levels of faults under electrical faults are represented by different symbols with the same red color. The levels of faults under mechanical faults are represented by different symbols with the same green color. The training data with mechanical faults are located in the grey region as shown in Fig. 4 (a). And they are far from the classification boundary. The training data with mechanical faults are well separated from the data with normal condition. This is reasonable because a minor mechanical fault can cause vibration signals changed greatly. Thus the data with mechanical faults are distant from the data with normal condition in the input space. For the training data with normal conditions, they all lie in the white region shown in Fig. 4(a). This means all the training data with normal condition are visualized in the correct region representing normal conditions. But for the data with 10Ω electrical fault (red circle), 7 of them lie in the white region representing the normal condition. This is reasonable because 10Ω electrical fault can be considered as a very small drift from normal condition. The vibration signals change slightly and are very close to the data with normal condition in the input space and in the reduced space. The other data with electrical faults are little far from the data with normal condition. The heavier the electrical faults are, the more distant they are from the normal data. If we can classify data from the visualization shown in Fig. 4(a), the classification accuracy is 97.7%, which is higher than the SVM classification accuracy of 97.0%. This is because two

data with normal condition are classified as faulty by SVM and as normal by SVMV while the other 7 data with faults are both mis-classified by SVM and SVMV. If the data are close to the classification boundary, they have high possibility of being mis-classified by SVMV. It is noted that the purpose of SVMV is visualization, and visual classification is a byproduct only. Similar results were obtained with the testing data. In Fig. 4 (b), the data with mechanical faults lie far from the data with normal condition. The distances between the normal and electrical faulty data become more distant in accordance with the increased of faulty level. The visual classification is 96.7%, which is a little higher than the actual classification accuracy (96.0%).

For ViSOM-based SVMV, λ in ViSOM is selected as 0.47 by trial and error. The other parameters of ViSOM are the same as those of SOM. The visualization results for the training and testing data are shown in Fig. 4 (c) and (d) respectively. Compared with Fig. 4 (a) and (b), the outside data boundary is clearly shown in Fig. 4 (c) and (d) by ViSOM while that is clipped in Fig. 4 (a) and (b) by SOM. The visualization results of normal, electrical faulty and mechanical faulty data are similar to those in SOM-based SVMV. With the same number of neurons and good visualization effect, ViSOM-based SVMV has 95.3% visual classification accuracy for the training data and 94.0% for the testing data. This is because the maps of SOM in Fig.4 (a) and (b) are a little more precise than those in Fig. 4 (c) and (d). Several traditional visualization methods, i.e., PCA, MDS, Sammon's mapping, CCA, ISOMAP, KPCA, and LLE are used. The visualization results by PCA and MDS are shown in Fig. 4(e) to (f) respectively. The visualization results by other methods are similar to Fig.4 (e) and (f). Similar to Fig. 4 (a) to (d), the normal and mechanical faulty data are well separated. And the normal data are mixed up by the light electrical faulty data. Although the classification boundary can be perceived by user, they are merely and can never be used as a reliable visual classification.

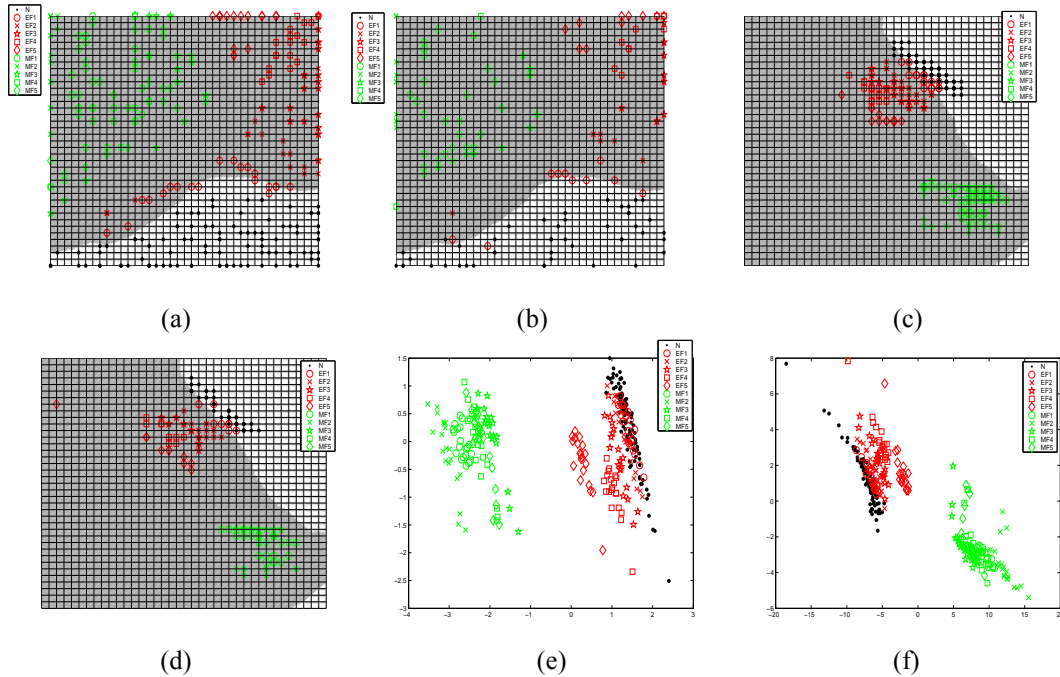


Fig. 4. Visualization results (a) by SOM-based SVMV on the training data; (b) by SOM-based SVMV on the testing data; (c) by ViSOM-based SVMV on the training data; (d) by ViSOM-based SVMV on the testing data; (e) by PCA on the training data; (f) MDS on the training data. “N” represents normal data. “EF1” to “EF5” represent electrical faults from light to heavy levels. “MF1” to “MF5” represent mechanical faults from light to heavy levels.

IV. CONCLUSION

Classical NN type method cannot provide good explanation for induction machine fault detection. In this paper, a visualization method is introduced to provide useful information to non-NN expert in understanding the classification process. Classical visualization methods do not utilize class information during the course of computation. It is hard to virtually impossible to give the class boundary among classes. In the proposed SVMV method, both SVM and SOM are used. It achieves good visualization effect with classification boundary. In the propose approach, the whole classification mechanism can be elaborated through the SVMV. To minimize mis-classification, one can simply perform the classification from the SVMV result. It is more accurate and robust. Experimental results on induction machine fault

detection demonstrate the novelty and significance of SVMV. Apparently, SVMV can become a very versatile tool for induction machine fault detection and other types of classifications.

REFERENCES

- [1] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods-An introduction," *Contr. Eng. Practice*, vol. 5, no. 5, pp. 639-652, 1997.
- [2] S. Leohardt and M. Ayoubi, "Methods of fault diagnosis," *Contr. 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] Z. Ye and B. Wu, "A Review on Induction Motor Online Fault Diagnosis," *The Third International Power Electronics and Motion Control Conference*, pp. 1353-1358, 2000.
- [5] M.-Y. Chow, *Methodologies of Using Neural Network and Fuzzy Logic for Motor Incipient Fault Detection*. Singapore: World Scientific, 1997.
- [6] F. Filippetti, G. Franceschini, C. Tassoni, P. Vas, "Recent developments of induction motor drives fault diagnosis using AI techniques," *IEEE Trans. Industrial electronics*, vol. 47, no. 5, pp. 994-1004, 2000.
- [7] M. -Y. Chow, R. N. Sharpe, and J. C. Hung, "On the application and design consideration of artificial neural network fault detectors," *IEEE Trans. Ind. Electron.*, vol. 40, no. 2, pp. 181-198, 1993.
- [8] M. -Y. Chow and C. Mangum, "A neural network approach to real time condition monitoring of induction machines," *IEEE Trans. Ind. Electron.*, vol. 38, no. 6., pp. 449-454, 1991.
- [9] I. E. Alguindigue, A. L. Buczak, and R. E. Uhrig, "Monitoring and diagnosis of rolling element bearings using artificial neural networks," *IEEE Trans. Industrial electronics*, vol. 40, no. 2, pp. 209-217, 1993.
- [10] F. Filippetti, G. Franceschini, and C. Tassoni, "Neural network aided on-line diagnostics of induction motor rotor faults," *IEEE Trans. Ind. Applicat.*, vol. 31, no. 4, pp. 892-899, 1995.
- [11] B. Li, M. Y. Chow, Y. Tipsuwan, "Neural-network-based motor rolling bearing fault diagnosis," *IEEE Trans. Industrial electronics*, vol. 47, no. 5, pp. 1060-1069, 2000.
- [12] S. Wu, T. W. S. Chow, "Induction machine fault detection: using SOM-based RBF neural networks," *IEEE trans. Industrial Electronics*, vol. 51, no. 1, pp. 183-194, 2004.
- [13] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.
- [14] J. E. Jackson, *A User's Guide to Principal Components*. New York: John Wiley & Sons, 1991.
- [15] T. Cox and M. Cox, *Multidimensional Scaling*. London: Chapman & Hall, 1994.
- [16] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer, 1997.
- [17] K. Diamantaras and S. Kung, *Principal Component NN Theory and Applications*. New York: John Wiley & Sons, 1996.
- [18] B. Schölkopf, A. Smola, and K. Müller, "Non-linear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.
- [19] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Trans. On Computers*, vol. 18, pp. 401-409, 1969.
- [20] P. Demartines and J. Hérault, "Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of datasets," *IEEE Trans. on Neural Networks*, vol. 8, pp. 148-154, 1997.
- [21] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319-2323, 2000.
- [22] S. T. Roweis and L. K. Saul, "Nonlinear dimensional reduction by locally linear embedding," *Sci*, vol. 290, pp. 2323-2326, 2000.
- [23] S. Wu, T. W. S. Chow, "Support vector visualization and clustering using self-organizing map and support vector one-class classification," in *Proc. of Int Joint Conference on Neural Networks (IJCNN)*, Portland, USA, August 2003, pp. 803-808.
- [24] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett, "New Support Vector Algorithms," *Neural Computation*, vol. 12, pp. 1207-1245, 2000.
- [25] H. Yin, "ViSOM-A novel method for multivariate data projection and structure visualization," *IEEE Trans. Neural Networks*, vol. 13, no. 1, pp. 237-243, 2002.
- [26] D. W. Thomas, *Vehicle sound recognition, a pilot study*. Ph.D. Thesis, University of Southampton, England, 1971.