COMPARISON OF RECONSTRUCTION SCHEMES OF MULTIPLE SVM’S APPLIED TO FAULT CLASSIFICATION OF A CAGE INDUCTION MOTOR

Sanna Pöyhönen  Marian Negrea  Antero Arkkio  Heikki Hyötyniemi
COMPARISON OF RECONSTRUCTION SCHEMES OF MULTIPLE SVM’S APPLIED TO FAULT CLASSIFICATION OF A CAGE INDUCTION MOTOR

Sanna Pöyhönen¹  Marian Negrea² Antero Arkkio² Heikki Hyötyniemi¹

¹Helsinki University of Technology
Control Engineering Laboratory

²Helsinki University of Technology
Laboratory of Electromechanics

Abstract: Different schemes to reconstruct a multi-class classifier from one-to-one support vector machine (SVM) based classifiers are compared with application to fault diagnostics of a cage induction motor. Power spectrum estimates of circulating currents in parallel branches of the motor are calculated with Welch’s method, and SVM’s are trained to distinguish healthy spectrum from faulty spectra and faulty spectra from each other. Majority voting, a mixture matrix and neural network are compared in reconstruction the global classification decision from outputs of SVM’s.

Keywords: support vector machine, multi-class classification, fault diagnostics, electrical machine, finite element analysis

Helsinki University of Technology
Department of Automation and Systems Technology
Control Engineering Laboratory
1. Introduction

Support vector machine (SVM) is a relatively new computational learning method based on statistical learning theory presented by V.N. Vapnik [1]. In SVM, original input space is mapped into a high dimensional dot product space called feature space, and in the feature space the optimal hyperplane is determined to maximize the generalisation ability of the classifier. The optimal hyperplane is found by exploiting a branch of mathematics called optimisation theory, and respecting insights provided by the statistical learning theory [2].

SVM’s are 2-class classifiers. They are designed to separate only two classes from each other. However, in most of the real applications multi-class classification is needed. One of the simplest multi-class classification structures is a so called one-against-others approach. In this method, $K$ pairwise classifiers are built in the way that each classifier separates one class from all the others. However, in many applications, this approach has been found to be inferior to a pairwise coupling approach, where $\frac{1}{2}K(K-1)$ pairwise classifiers are built, each separating one class from another ignoring all the other classes. Pairwise classifiers’ outputs are then fused to get the final multi-class classification decision.

There exist several ways to do the fusion of pairwise classifiers’ outputs. In this paper, we study four different methods chosen because of their clarity and intuitiveness. Firstly, majority voting schemes with soft and rough reconstructions are studied. Secondly, we study a mixture matrix that linearly combines pairwise classifiers. Then nonlinear combination is studied in the form of a neural network. The voting scheme is proposed in [3], and it is also studied in [4]. In [5], a mixture matrix approach is proposed, although it has been considered there with one-against-others type of classification.

In [6] and [7], more complicated approaches are proposed: in [6], binary trees of SVM’s are considered to solve a multi-class pattern recognition problem, and in [7], fuzzy logic is proposed to combine outputs of SVM’s. Binary trees are left out from this study, because when applying them, a proper hierarchy of classifiers should be known before training the classifiers. This requires a priori knowledge of the solution of the classification problem or implementation of sophisticated clustering or vector quantisation algorithms. Also the fuzzy logic approach is left out from this study, because choosing and tuning the membership functions are time-consuming and application dependent tasks.

Comparison of methods is done in application to fault diagnostics of a cage induction motor. Rotating electrical machines play a very important role in the world's industrial life, and, thus, there is a strong industrial demand on their reliable and safe operation. Faults and failures of critical electro-mechanical parts can lead to excessive downtimes and generate costs of millions of euros in reduced output, emergency maintenance and
lost revenues. In addition to traditional model based methods and expert systems, numerical classification methods have gathered following in the area of modern fault diagnostics of electrical machines. For example, in [8,9,10], neural network based classification has been applied to monitor rolling bearings of motor. We have studied SVM based classification for fault diagnostics of an electrical machine in [11, 12]. In [11], we constructed pairwise SVM’s to separate faults of the machine from healthy operation and faulty situations from each other based on spectral information of a stator line current of a motor. In [12], we constructed a SVM based multi-class classifier for fault diagnostics of several faults. A rough majority voting approach was used in combination of pairwise classifiers. In this paper, we compare different combination approaches, and instead of the line current, the classification of faults is based on spectral information of circulating currents in parallel branches of the motor.

The content of the paper is following: in Section 2, SVM theory is presented based on [1] [2] and [13], in Section 3, multi-class classification is clarified with different reconstruction schemes, in Section 4, numerical simulations of a cage induction motor’s variables are described, in Section 5, the power spectrum estimation of the motor current is considered, in Section 6, the classification results are presented, and in Section 7, some conclusions are done.
2. SVM based classification

Let $n$-dimensional input $\mathbf{x}_i$ ($i = 1,\ldots,M$) belong to Class I or Class II and associated labels be $y_i = 1$ for Class I and $y_i = -1$ for Class II. For linearly separable data, we can determine a hyperplane $f(x)$ that separates the data:

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b = \sum_{j=1}^{n} w_j x_j + b .$$  (1)

A separating hyperplane satisfies the constraints that define the separation of the data samples, i.e. $f(x_i) \geq +1$, if $y_i = +1$, and $f(x_i) \leq -1$, if $y_i = -1$ [13, p. 357]. This results:

$$y_i f(x_i) = y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 , \text{ for } i = 1,\ldots,M .$$  (2)

where $\mathbf{w}$ is an $n$-dimensional vector and $b$ is a scalar. Notation $\mathbf{w} \cdot \mathbf{x}_i$ corresponds to dot product of vectors $\mathbf{w}$ and $\mathbf{x}_i$. The weighting vector $\mathbf{w}$ defines the direction of the separating hyperplane $f(x)$ and with $\mathbf{w}$ and $b$ (bias) it is possible to define the hyperplane’s distance from the origin.

The separating hyperplane that has the maximum distance between the hyperplane and the nearest data, i.e. the maximum margin, is called the optimal separating hyperplane. An example of optimal separating hyperplane of two datasets is presented in Fig. 1. The optimal hyperplane is perpendicular to the shortest line between border lines of two sets, and the plane and the shortest line intersect each other in the halfway of the line.

The geometrical margin $\gamma$ is half of the sum of the distances between arbitrary separating hyperplane and the nearest negative and positive datum ($\mathbf{x}^-$ and $\mathbf{x}^+$):

$$\gamma = \frac{1}{2} \frac{((\mathbf{w} \cdot \mathbf{x}^-) - (\mathbf{w} \cdot \mathbf{x}^+))}{\|\mathbf{w}\|_2} = \frac{1}{2} \frac{((\mathbf{w} \cdot \mathbf{x}^+) - (\mathbf{w} \cdot \mathbf{x}^-))}{\|\mathbf{w}\|_2} .$$

Without loss of generality we can search the optimal separating hyperplane among so called canonical hyperplanes, which fulfil $\mathbf{w} \cdot \mathbf{x}^+ + b = 1$ and $\mathbf{w} \cdot \mathbf{x}^- + b = -1$ [2]:

$$\gamma = \frac{1}{2} \frac{((\mathbf{w} \cdot \mathbf{x}^+) - (\mathbf{w} \cdot \mathbf{x}^-))}{\|\mathbf{w}\|_2} = \frac{1}{\|\mathbf{w}\|_2} .$$

The optimal hyperplane maximizes the geometrical margin. Thus the optimal hyperplane can be found by solving the following convex quadratic optimisation problem:

minimize $\frac{1}{2} \|\mathbf{w}\|^2$
subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 .$  (3)
The same optimisation problem can also be formulated by minimizing the guaranteed risk for classification problem (i.e. maximizing the generalisation ability). For this approach, see e.g. [2].

If the number of attributes of data examples is large, we can considerably simplify calculations by converting the problem with Kuhn-Tucker conditions into the equivalent Lagrange dual problem. Lagrange function for (3) is:

$$L(w, b, \alpha) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^{M} \alpha_i \left[ y_i \left( (w_i \cdot x_i) + b \right) - 1 \right], \quad (4)$$

where $\alpha = (\alpha_1, \ldots, \alpha_M)$ is the Lagrange multiplier. The dual problem is:

$$\begin{align*}
\text{maximize} & \quad L(w, b, \alpha) \\
\text{subject to} & \quad \alpha_i \geq 0, \quad i = 1,\ldots,M \quad (5)
\end{align*}$$

By differentiating (4) with respect to $w$ and $b$ and imposing stationarity, we get:

$$\begin{align*}
\frac{\partial L}{\partial w}(w, b, \alpha) &= w - \sum_{i=1}^{M} y_i \alpha_i x_i = 0 \\
\frac{\partial L}{\partial b}(w, b, \alpha) &= \sum_{i=1}^{M} y_i \alpha_i = 0 \quad (6)
\end{align*}$$

From (4), (5) and (6) we get the dual representation of the optimisation problem:

$$\begin{align*}
\text{maximize} & \quad W(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,k=0}^{M} \alpha_i \alpha_k y_i y_k x_i \cdot x_k \\
\text{subject to} & \quad \sum_{i=1}^{M} y_i \alpha_i = 0, \quad \alpha_i \geq 0, \quad i = 1,\ldots,M \quad (7)
\end{align*}$$
The number of variables of the dual problem is the number of training data.

Let us assume that optimal solution for the dual problem is $\alpha^*$ and $b^*$. According to the Karush-Kuhn-Tucker theorem, the equality condition in (2) holds for the training input-output pair $(x_i, y_i)$ only if the associated $\alpha_i^*$ is not 0. In this case the training example $x_i$ is a support vector. Solving (7) for $\alpha = (\alpha_1, \ldots, \alpha_M)$, we can obtain the support vectors for classes 1 and 2. Then the optimal separating hyperplane is placed at the equal distances from the support vectors for classes 1 and 2, and $b^*$ is given by:

$$
 b^* = -\frac{1}{2} \sum_{k=1}^{M} y_k \alpha_k^* (s_1 \cdot x_k + s_2 \cdot x_k),
$$

where $s_1$ and $s_2$ are respectively, arbitrary support vectors for Class I and Class II. In Fig. 1, support vectors are bolded. Notice that support vectors are such training samples that are on the margin of two datasets. The optimal separating hyperplane would be the same, if only support vectors had been used as training data.

So far we have assumed that the training data is linearly separable. In the case where the training data cannot be linearly separated, we introduce non-negative slack variables $\xi_i$ to (2), and add to the objective function given by (5), the sum of the slack variables multiplied by the parameter $C$. This corresponds to adding the upper bound $C$ to $\alpha$. In both cases, the decision functions are the same and are given by:

$$
 f(x) = \sum_{i=1}^{M} \alpha_i^* y_i x_i \cdot x + b^* .
$$

Then unknown data example $x$ is classified as follows:

$$
 x \in \begin{cases} 
 \text{Class I, if } f(x) \geq 1 \\
 \text{Class II, if } f(x) < -1 
\end{cases}.
$$

SVM is a non-linear kernel-based classifier, which maps the data to be classified onto a space, where the data can be linearly classified. The space is called a feature space. Using the non-linear vector function $\Phi(x) = (\Phi_1(x), \ldots, \Phi_l(x))$ that maps the $n$-dimensional input vector $x$ into the $l$-dimensional feature space, the linear decision function in dual form is given by

$$
 f(x) = \sum_{i=1}^{M} \alpha_i y_i \Phi(x_i) \cdot \Phi(x). 
$$

(8)

Notice that in (8) as well in the optimisation problem (7), the data occur only in inner products. In SVM, the actual mapping function, $\Phi$, is not necessary to be known, but the classes optimally separating hyperplane is possible to calculate with inner products of the original data samples. If it is possible to find this kind of procedure to calculate inner products of feature space in original data space, it is called a kernel, $K(x, z) = \Phi(x) \cdot \Phi(z)$. Then the learning in the feature space does not require
evaluating \( \Phi \) or even knowing it, because all the original samples are handled only with Gram matrices \( G = (x_i \cdot x_j)^{M_{i,j}} \). Using a Kernel function, the decision function will be:

\[
f(x) = \sum_{\text{support vectors}} \alpha_i^* y_i K(x_i, x).
\]

However, not all kernels correspond to inner products in some feature space. With a so-called Mercer’s theorem it is possible to find out whether a kernel \( K \) depicts an inner product in that space where \( \Phi \) is mapped [2]. For example, polynomials of degree \( q \) have inner product kernel

\[
K(x, z) = (x \cdot z + 1)^q
\]

and radial basis functions of the form

\[
\varphi(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i \exp \left\{ -\frac{|x - x_i|^2}{\sigma^2} \right\} \right), \text{ where } \sigma \text{ defines the width, have the inner product kernel}
\]

\[
K(x, z) = \exp \left\{ -\frac{|x - z|^2}{\sigma^2} \right\}.
\]
3. Multi-class classification

In many real classification applications, a method to deal with several classes is required. For example, in fault diagnostics of an electrical machine, there exist several fault classes in addition to healthy operation. SVM is a 2-class classifier, but it is possible to decompose a multi-class problem to several 2-class problems, train pairwise classifiers to solve these 2-class problems, and then reconstruct the solution of the multi-class problem from outputs of classifiers. For a \( K \)-class classification problem, \( \frac{1}{2} K(K-1) \) pairwise classifiers cover all pairs of classes. Each classifier is trained on a subset of the training set containing only training examples of the two involved classes. To find the global solution to the \( K \)-class problem there exist several methods to reconstruct the solutions of 2-class problems. In this chapter, four reconstruction schemes are considered. All considered reconstruction schemes can be used with other pairwise classifiers in addition to SVM.

3.1 Majority voting

When an input vector \( \mathbf{x} \) is to be classified, it is presented to all pairwise classifiers, each providing a partial answer that concerns the two involved classes. An intuitive approach to get the global classification result is to consider outputs of the classifiers as votes and select the class that gets most votes. In [4], this is formulated in a following way.

Assuming that the classifier \( f_{ij}(\mathbf{x}) \) discriminating between class \( i \) and \( j \) computes an estimate \( \hat{p}_{ij} \) of the probability \( p_{ij} = P(\mathbf{x} \in \text{class}_i | \mathbf{x} \in \text{class}_i \cup \text{class}_j) \), the final classification decision is determined by \( \hat{F} = \arg \max_{1 \leq i \leq K} \sum_{j \neq i} \left[ \hat{p}_{ij} > 0.5 \right] \). Here the operator \( [\cdot] \) is defined as:

\[
[\eta] = \begin{cases} 
1, & \text{if } \eta \text{ is true} \\
0, & \text{otherwise.} 
\end{cases}
\]

This combination considers the outputs of the classifiers as binary decisions. In [4], another reconstruction scheme is also formulated assuming that the outputs \( \hat{p}_{ij} \) of the classifiers can be used to calculate approximations \( \hat{p}_i \) of the a posteriori probabilities \( p_i = P(\mathbf{x} \in \text{class}_i) \). Considering a square matrix \( \hat{\mathbf{P}} \) with the value \( \hat{p}_{ij} \) in position \((i, j)\), and with \( \hat{p}_{ij} = 1 - \hat{p}_{ji} \), the values of \( \hat{p}_i \) can be obtained from:

\[
\hat{p}_i = \frac{2}{K(K-1)} \sum_{j \neq i} \hat{p}_{ij} ,
\]

(9)
and the final classification decision is made with $\hat{F} = \arg\max_{i \in \mathbb{K}} \hat{p}_i$. According to [4], this approach is called *soft reconstruction* and the first approach *rough reconstruction*.

When applying SVM classifiers, the outputs of the classifiers cannot be interpreted as pure probabilities, like in [4]. However, the bigger is the output of a SVM classifier with a specific sample to be classified, the more likely the sample belongs to the positive class, and correspondingly, the smaller is the output, the more likely the sample belongs to the negative class. Thus, both majority voting schemes should be appropriate also for reconstructing the global solution from SVM solutions of decomposed pairwise classification problems.

### 3.2 Mixture matrix reconstruction

An important problem occurs when applying either of the previous methods. For a given sample $x$, the vote counting takes into account similarly the outputs of all pairwise classifiers, without considering their significance. One does not know in advance which classifiers give important knowledge of a certain sample – that is what one tries to find out by classification. However, redundancy of some pairwise classifiers may be considered with a so called *mixture matrix*. With this approach, outputs of classifiers are linearly combined with the mixture matrix created, for example with least square estimation, to minimize the error between the correct class decision and the linear combination of outputs of pairwise classifiers. Compared to the majority voting approach, this approach weights the votes given by classifiers so that the classification is more likely correct in the least square sense. Unlike the previous reconstruction schemes, a mixture matrix approach should be able to take into account possible malfunction of some pairwise classifiers.

Say, we have $n = \frac{1}{2}K(K-1)$ pairwise classifiers and their outputs are $g_n = (r_1, r_2, \ldots, r_n)^T$. $M$ is a $K \times n$ matrix, and it is estimated after training the classifiers to emphasize outputs of classifiers that have strong correlation with a correct classification decision. We have used here a regular multilinear regression (MLR) to find $M$. Also, partial least square method (PLS) or principal component regression (PCR) could be utilised in this task.

Fusion of the classifiers is done so that the mixed output $g_{\text{mixed}} = Mg_n$ is a $K \times 1$ sized vector, whose elements are in ideal situation all zeros except the one that corresponds to the class the considered sample represents. However, elements are not exactly zeros and ones due to noise, and the maximum of the elements shows the global classification decision:

$$\hat{F} = \arg\max_{i \in \mathbb{K}} g_{\text{mixed}}$$.
3.3 Neural network reconstruction

In the mixture matrix approach, a linear combination of pairwise classifiers outputs is calculated to make the final class decision. Instead of linear combination, also nonlinear combination can be utilised. As a nonlinear approach we study here a neural network that is trained to minimize the error between the correct class decision and the nonlinear combination of outputs of pairwise classifiers. Similarly with the previous approach, the neural network has $n$ inputs and $K$ outputs, and the maximum of outputs shows the global classification decision.

We use a regular feed-forward network with log-sigmoid transfer function. The log-sigmoid transfer function was chosen because its output range (0 to 1) suits to output Boolean values. Training is done using backpropagation with both adaptive learning rate and momentum.
4. Providing virtual measurement data from a 35 kW cage induction motor

The simulation of an electrical machine is based on time-stepping, finite-element analysis. The magnetic field in the core region of the machine is assumed to be two-dimensional. The three-dimensional end-region fields are modeled approximately using end-winding impedances in the circuit equations of the windings. The field and circuit equations are discretised and solved together. The rotation of the rotor is achieved by changing the finite-element mesh in the air gap, and the time-dependence is modelled by the Crank-Nicholson method. The magnetic field, the currents and the potential differences of the windings are obtained in the solution of the coupled field and circuit equations. Most of the other machine characteristics can be derived from these quantities. Many other physical parameters can be computed from these variables. The details of the method have been presented by Arkkio in [14].

The circuit equations of the stator and rotor windings are modified to implement the faults (Table 1). A shorted coil in stator winding is obtained by extracting a coil from the series connected coils forming a healthy phase and forcing the voltage of the extracted coil to be zero. A shorted turn is constructed in a similar manner. A broken bar or end-ring in the rotor cage is obtained by adding a large resistance properly in the circuit equations of the rotor. Static eccentricity is obtained by shifting the rotor by 10% of the radial air-gap length, and rotating the rotor around its center point in this new position. Dynamic eccentricity is obtained by shifting the rotor by 10% of the air-gap length, but rotating it around the point that is the center point of the stator bore. A 10% eccentricity is not yet a real fault but an eccentricity possibly growing should be detected at an early stage.

| NF | Healthy Machine |
| BB | Broken Rotor Bar |
| BR | Broken End-Ring in Rotor Cage |
| SC | Shorted Coil in Stator Winding |
| ST | Shorted Turn in Stator Winding |
| SE | Static Eccentricity in Rotor |
| DE | Dynamic Eccentricity in Rotor |

The magnetic field of a healthy electrical machine is usually periodic from pole to pole, and it is often enough to model one or two pole pitches. A fault in the machine disturbs the geometric symmetry, and the whole cross section has to be modelled. This may lead to large problems having tens of thousands of unknown variables. Fortunately, we are more interested in qualitative than exact quantitative results, and the finite element meshes used for the purposes of fault detection can be relatively sparse, as long as the geometric symmetry of the faulted machine is preserved.
The studied machine is an inverter-fed 35 kW squirrel cage induction motor with a star connection of the stator winding. Three different load conditions are considered: no load, half load, full load. Fig. 2 shows the cross-sectional geometry of the motor.

Figure 2. Contour lines of the magnetic vector potential on a quadrant of the 35 kW cage induction motor. In fault simulations, the whole cross-section of the machine has to be modelled.
5. Power spectra estimation of circulating currents

Power spectra estimates of the stator line current have often been used as a medium of fault detection of electrical machines [15]. Instead of the line current, we use here circulating currents in parallel branches as the basis of classification, because in our studies it has shown to be highly superior indicator of faults compared to the line current [16]. Circulating currents are close to zero in healthy operation of the motor and increase in fault situations. Using the line current as a medium of fault detection is advisable only because measuring it does not require access to the motor.

Main disadvantage of classical spectral estimation techniques, like FFT, is the impact of side lobe leakage due to windowing of finite data sets. Window weighting decreases the effect of side lobes. Further, in order to improve statistical stability of the spectral estimate, averaging by segmenting the data can be applied. The more segments are used the more stable the estimate is. However, the signal length limits the number of segments used, but with overlapping segments the number of segments can be increased. In this paper, Welch’s method [17] is used to calculate the power spectra estimates of circulating currents of the induction motor. The method applies both the window weighting and the averaging over overlapping segments to estimate the power spectrum. In this study, we use Hanning window sized 500 samples, and number of overlapping samples is 250.

Before applying power spectrum estimation measurement noise is imitated by adding normally distributed noise to the current. Mean value of noise is equal to zero and its standard deviation is 5% of the amplitude of current.
6. Results

6.1 Generating the sample set

The power spectra estimates were calculated 32 times from different parts of a circulating current signal in each fault case. Thus, we got a spectrum sample set with 160 samples for three different load cases. The classification structure was trained and tested separately in different load situations. Half of the samples were chosen for training the classifier, and half of the samples were left for testing the classifier’s generalisation ability. An average healthy spectrum from the training set was chosen to be a reference, and all the other spectra were scaled with it. The difference values from the reference created the sample set. As an example, the sample set of shorted coil fault is plotted in no load situation in Fig. 3. In the first picture, power spectrum estimates of a current in healthy operation and in shorted coil operation are plotted. The spectra are scaled with an inverse of the sampling frequency 20 kHz. In the second picture, the difference between the average healthy spectrum and shorted coil spectra are plotted.

Different faults have different kinds of impacts on the power spectra estimate of circulating currents. For comparison to a shorted coil case, a sample set of the power spectrum estimate of the current in static eccentricity operation is shown in Fig. 4. Shorted coil fault has a slightly stronger impact on the circulating current spectrum than static eccentricity, but also in static eccentricity operation the spectra samples differ from healthy operation spectra. Especially, in low frequencies, it is possible to detect peaks in the difference spectra on fault dependent frequencies.

Figure 3. Sample set of Welch’s power spectra estimates of a circulating currents in healthy and in shorted coil operation.
6.2 Performance measure

When 2-class classifiers are considered, true positive rates (TPR) and false positive rates (FPR) are intuitive choices for measuring performance of the classifier. The same measures are often used when a one-against-others type of classification is applied to a multi-class problem. However, when a multi-class problem is solved with a pairwise classification scheme, there does not exist such concepts as true positives or true negatives, because solution to the problem is formed based on outputs of several classifiers. For example, there may be situations, where an individual sample is classified to be positive for several classes. This can especially happen when using a majority voting approach with rough reconstruction.

Here we will use a standard $Q$ percentage accuracy to evaluate performance of the classification structure (used e.g. in [18]). Suppose we have $n_1 + n_2 + \ldots + n_K$ test samples from $K$ classes ($n_i$ are observed to belong to class 1, etc.). Suppose that out of $n_i$ samples, $q_i$ are correctly and uniquely classified. Then the accuracy for class $i$ is $Q_i = q_i/n_i$. The overall accuracy is $Q_{tot} = \sum_i Q_i$. Here “unique” means that a test sample $j$ is classified to only one class, i.e. its correctness measure $c_j = 1$. If a sample $j$ is classified to be positive for more than one class, then its correctness measure is less than one. For example, if a sample $j$ is classified to be positive for 3 classes, then $c_j = 1/3$ for this sample. So $q_i = \sum_j c_j$ is not necessarily an integer.
6.3 Classification results

Choosing a kernel function for SVM’s has a considerable impact on classification results. However, there does not exist practical rules for choosing the kernel function, the best kernel function depends on the classification problem considered. Choosing the kernel function and the parameter $C$ is a difficult task in a multi-classification problem, because all SVM’s may have individual best choices for them concerning the success of the whole classification, and all the possible combinations of the parameters should be tested. In this study, all pairwise classifiers were designed with a same simple inner product kernel function corresponding to a first order polynomial in the feature space. Practically, similar results we achieved with a kernel function corresponding a properly parametrized radial basis function and higher order polynomials, whose degree was an odd number. The upper bound $C$ for the Lagrange multipliers was chosen to be 1.

When using a neural network approach in combination of classifiers, we first had to consider the construction of the network. The number of neurons in a hidden layer has a strong impact on the performance of the network. With a large number of hidden neurons, it is possible to achieve excellent performance in the training set, but this does not necessarily lead to good generalisation ability and high accuracy in the evaluation set. After numerous tests, we chose the number of hidden neurons that gave the best overall accuracy in the evaluation set. For no load classification structure the number was 12, for half load classification structure 18 and for full load classification structure 8. The classification results with these structures are presented in Table 2. In the table, first six columns represent the accuracies $Q_i$ for each class, and the overall accuracy $Q$ is calculated in the last column for all load cases. In the last row, the total accuracies over all load cases are presented.

In Table 3, correct classification percentages are presented, when majority voting with rough reconstruction is used. In Table 4, classification results are presented, when soft reconstruction is used. In Table 5, there are presented results with a mixture matrix approach.

<table>
<thead>
<tr>
<th>%</th>
<th>$Q_i$ (NF)</th>
<th>$Q_i$ (BB)</th>
<th>$Q_i$ (BR)</th>
<th>$Q_i$ (ST)</th>
<th>$Q_i$ (SC)</th>
<th>$Q_i$ (SE)</th>
<th>$Q_i$ (DE)</th>
<th>$Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Load</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Half Load</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>No Load</td>
<td>57.5</td>
<td>57.5</td>
<td>97.5</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>87.5</td>
</tr>
<tr>
<td>All</td>
<td>85.8</td>
<td>85.8</td>
<td>99.2</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>95.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>%</th>
<th>$Q_i$ (NF)</th>
<th>$Q_i$ (BB)</th>
<th>$Q_i$ (BR)</th>
<th>$Q_i$ (ST)</th>
<th>$Q_i$ (SC)</th>
<th>$Q_i$ (SE)</th>
<th>$Q_i$ (DE)</th>
<th>$Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Load</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>50.0</td>
<td>100</td>
<td>92.9</td>
</tr>
<tr>
<td>Half Load</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>50.0</td>
<td>100</td>
<td>92.9</td>
</tr>
<tr>
<td>No Load</td>
<td>47.5</td>
<td>45.8</td>
<td>92.5</td>
<td>100</td>
<td>100</td>
<td>52.5</td>
<td>100</td>
<td>76.9</td>
</tr>
<tr>
<td>All</td>
<td>82.5</td>
<td>81.9</td>
<td>97.5</td>
<td>100</td>
<td>100</td>
<td>50.8</td>
<td>100</td>
<td>87.6</td>
</tr>
</tbody>
</table>
The overall accuracies with different reconstruction schemes over all load situations are gathered in Table 6. The best results are gained with neural network reconstruction (95.8%) and the worst results with majority voting with rough reconstruction (87.6%). Majority voting with soft reconstruction (93.9%), mixture matrix reconstruction (95.5%) and neural network reconstruction were highly competitive compared to it. The difference between these combination methods can especially be seen in detection of static eccentricity faults. With majority voting and rough reconstruction, static eccentricity fault samples tend to get equal amount of votes with shorted turn fault class. Thus, the total accuracy of the classifier degrades.

In no load situation, problems occur also in detection of some other faults in addition to static eccentricity, and these problems cannot be totally solved by changing the reconstruction scheme. It is obvious that faults are more easily detected from a motor that is working under load. If differences in a faulty spectrum sample compared to the average healthy sample are hidden due to noise, classification does not succeed.

The neural network approach and the mixture matrix approach resulted in a classification structure with almost same accuracy. When comparing the results, it should be taken into account that with arbitrarily chosen number of hidden neurons, the neural network could result in much worse classification results. Also, training and tuning the neural network is an exhausting task. A linear combination of SVM’s with a mixture matrix already gives adequate results, so using nonlinear reconstruction is not necessary in this application.
7. Conclusions

Different schemes to reconstruct a multi-class classifier from one-to-one SVM based classifiers were compared with application to fault diagnostics of a cage induction motor. A neural network reconstruction resulted in the best multi-class classification structure, but with a reconstruction approach relying on a mixture matrix almost equal classification performance was obtained. In this application, a linear mixture matrix is a practical choice for the reconstruction scheme, because training and tuning a neural network is an exhausting task, and the benefits of applying a nonlinear approach are marginal. Majority voting approach with rough reconstruction was found to be inferior to the other methods considered.
References

[3]: Friedman, J., “Another Approach to Polychotomous Classification”, Technical Report, Stanford University, Department of Statistics, 1996


Report 118  Nissinen, A. S.

Report 119  Gadoura, I. A.

Report 120  Ylöstalo, T., Salonen, K., Siika-aho, M., Suni, S., Hyötyniemi, H., Rauhala, H., Koivo, H.

Report 121  Cavazzutti, M.

Report 122  Uykan, Z.

Report 123  Jäntti, R.

Report 124  Uykan, Z.

Report 125  Hyötyniemi, H.

Report 126  Kaartinen, J.

Report 127  Ylen, J.-P.

Report 128  Gadoura, I. A., Suntio, T.

Report 129  Elmusrati, M. S.

Report 130  Pöyhönen, S., Negrea, M., Arkkio, A., Hyötyniemi, H.