Detection of Short Circuit Faults in 3-Phase Converter-Fed Induction Motors Using Kernel SOMs

David N. Coelho, Guilherme A. Barreto
Federal University of Ceará - Campus of Pici
Department of Teleinformatics Engineering
Center of Technology, Fortaleza, Ceará, Brazil
Emails: davidcoelho89@gmail.com, gbarreto@ufc.br

Cláudio M. S. Medeiros
Federal Institute of Ceará - Campus of Fortaleza
Industry Department, Fortaleza, Ceará, Brazil
Email: claudiosa@ifce.edu.br

***Abstract*—In this work we report the results of a comprehensive study involving the application of kernel self-organizing maps (KSOM) for early detection of interturn short-circuit faults in a three-phase converter-fed induction motor. For this purpose, two paradigms for developing KSOM-based classifiers are evaluated on the problem of interest, namely the gradient descent based KSOM (GD-KSOM) and the energy function based KSOM (EF-KSOM). Their performances are contrasted on a real-world dataset generated by means of a laboratory scale testbed that allows the simulation of different levels of interturn short-circuits (high and low impedance) for different load conditions. Feature vectors are built from the FFT-based spectrum analysis of the stator current, a non-invasive method known as the stator *current signature*. The performances of the aforementioned KSOM paradigms are evaluated for different kernel functions and for different neuron labeling strategies. The obtained results are compared with those achieved by standard SOM-based classifier.**

I. Introduction

Induction motors comprise a class of simple, efficient and robust electric equipments with ubiquitous presence in modern industry [1], [2], [3]. As such, they are responsible for impressive numbers, for example, they correspond to approximately 40 to 50% of all the generated power capacity of an industrialized nation [2].

It is considered a very robust equipment because it can present acceptable performance even under very hazardous conditions, but machine aging, successive exposure to hard environmental conditions, inadequate use, and lack of preventive maintenance, eventually lead induction motors to present different types of faults [1], [4]. Failure rates are conservatively estimated as 3-5% per year [5], with the most common ones being related to bearing faults, stator or rotor insulation faults, open bars or crack of the rings, and eccentricity [4], [6].

Of particular interest to this work is the insulation breakdown in the stator winding, a type of fault that causes interturn short-circuit currents, and corresponds to nearly 40% of the total motor failures [4], [7]. Such a problem initially is characterized by a high impedance fault [8]; that is, short-circuit currents are of low intensity, making them difficult to detect at early stages. However, even initially small, short-circuit current gradually promotes local heating at the site of the insulation breakdown as time passes, causing the failure to spread quickly across the winding [9].

If this type of fault is detected at early stages, by means of a procedure called *Incipient Interturn Fault Diagnosis*, maintenance teams can act to avoid sudden stops of production lines and save production costs, minimizing also the damages to the induction machine, which can be reused after the motor rewinding [10]. Since the stator winding interturn short-circuit takes just few minutes to evolve [2] and due to all the impact an unscheduled production downtime can have in the whole involved supply chain, it is highly desirable to develop fault monitoring tools to reduce costs and increase the probability of saving the machines. Bearing this in mind, previous studies have advanced in this direction.

For example, in [11], neural network architectures and the wavelet transform are used in order to detect temporary short circuit in induction motor winding. In [12], the authors apply a two-dimensional Self-Organizing Map (SOM) [13] for detecting incipient short-circuit faults in an induction motor, achieving 88.63% as the best accuracy rate. In [14], a wavelet based probabilistic neural network is used for interturn fault detection. In [15], several pattern classifiers, including the Multi-layer Perceptron (MLP) [16], Extreme Learning Machine (ELM) [17], Support-Vector Machine (SVM) [18], Least-Squares Support-Vector Machine (LSSVM) [19], and the Minimal Learning Machine (MLM) [20], have their performances compared in the detection of incipient short circuit of an induction motor.

Motivated by the promising results reported in the aforementioned works, we present a comprehensive study involving the application of two kernelized variants of the SOM [21], [22], [23] for the detection of interturn short-circuit faults in a three-phase converter-fed induction motor. Their performances are contrasted with those achieved by the standard SOM-based classifier on a real-world dataset. A laboratory scale testbed allows the simulation of different levels of interturn short-circuits (high and low impedance) for different load conditions. The stator *current signature* method, which is a non-invasive FFT-based method for spectrum analysis of the stator current, is chosen as the feature extraction method. The performances of all classifiers are evaluated for different kernel functions and for different neuron labeling strategies.

The remainder of the paper is divided as follows. The SOM algorithm and neuron labeling strategies are briefly presented in Section II. kernel SOM algorithms are described in detail in Section III. In Section IV, we describe the laboratory scale experimental testbed mounted to generate the dataset to be

978-1-5090-6638-4/17/$31.00 ©2017 IEEE

used for training and testing of the classifiers. The obtained results are reported and discussed in Section V. The paper is concluded in Section VI.

II. Basics of the SOM Algorithm

The *Self-Organizing Map* (SOM) is an unsupervised competitive learning algorithm [13]. The SOM learns from examples a mapping (projection) from a high-dimensional continuous input space $\mathcal{X}$ onto a low-dimensional discrete space (lattice) $\mathcal{A}$ of Q neurons which are arranged in fixed topological forms, e.g., as a rectangular 2-dimensional array. The map $i^*(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{A}$, defined by the weight matrix $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_C\}, \mathbf{w}_i \in \mathbb{R}^p \subset \mathcal{X}$, assigns to each input vector $\mathbf{x}(n) \in \mathbb{R}^p \subset \mathcal{X}$ a winning neuron $i^*(n) \in \mathcal{A}$, determined by

$$i^*(n) = \arg\min_{\forall i} \|\mathbf{x}(n) - \mathbf{w}_i(n)\|^2, \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean distance and n symbolizes a discrete time step associated with the iterations of the algorithm.

The weight vector of the current winning neuron as well as the weight vectors of its neighboring neurons are simultaneously adjusted according to the following learning rule:

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta(n)h(i^*, i; n)[\mathbf{x}(n) - \mathbf{w}_i(n)] \quad (2)$$

where $0 < \eta(n) < 1$ is the learning rate and $h(i^*, i; n)$ is a weighting function which limits the neighborhood of the winning neuron. A usual choice for $h(i^*, i; n)$ is given by the Gaussian function:

$$h(i^*, i; n) = \exp\left(-\frac{\|\mathbf{r}_i(n) - \mathbf{r}_{i^*}(n)\|^2}{2\sigma^2(n)}\right) \quad (3)$$

where $\mathbf{r}_i(n)$ and $\mathbf{r}_{i^*}(n)$ are respectively, the coordinates of the neurons i and i^* in the output array, and $\sigma(k) > 0$ defines the radius of the neighborhood function at iteration n. The variables $\eta(n)$ and $\sigma(n)$ have to decay with time in order to guarantee convergence of the weight vectors to stable steady states. In this paper, we adopt a linear decay for both variables:

$$\eta(n) = \eta_0 \left(1 - \frac{n}{n_{max}}\right) \text{ and } \sigma(n) = \sigma_0 \left(1 - \frac{n}{n_{max}}\right), \quad (4)$$

where η_0 and σ_0 are the initial values of $\eta(n)$ and $\sigma(n)$), respectively; n_{max} is the maximum number of iterations.

Weight adjustment is performed until a steady state of global ordering of the weight vectors has been achieved. In this case, we say that the map has converged. The resulting map also preserves the topology of the input samples in the sense that adjacent patterns are mapped into adjacent regions on the map. Due to this topology-preserving property, the SOM is able to cluster input information and spatial relationships of the data on the map. Despite its simplicity, the SOM algorithm has been applied to a variety of complex problems [24], [25], [26] and has become one of the most popular artificial neural networks.

A. SOM Labeling Strategies

Since the SOM is an unsupervised learning algorithm, it requires post-training neuron labeling strategies in order to be applied to pattern classification problems. During the neuron labeling phase, training data samples are presented once more to the SOM, but the weights are not updated. Once this phase is concluded, the SOM becomes a prototype based classifier [27], [28]. Basically, all neuron labeling strategies aim at associating to each SOM neuron a class label, so that the nearest prototype rule in Eq. (1) is used for pattern classification purposes.

Three labeling methods are used in this paper, namely: the minimum distance, the average distance and the majority voting method. These methods are briefly discussed next.

- For the **minimum distance method**, the simplest one, the neuron inherits the label of its closest training sample.
- In the **average distance method**, one first needs to compute the distances from a given neuron to the samples of all classes which are mapped to this neuron. Then, compute the average distance per class. Finally, the neuron receives the label of the class whose associated average distance is the smallest one.
- In the **majority voting** method, each neuron inherits the most frequent class label among the labels of the samples mapped to it.

In the next section, we discuss two different paradigms for obtaining kernelized versions of the SOM algorithm.

III. Kernel SOM based classifiers

In recent years, kernel-based methods have been proposed with the aim of developing nonlinear versions of linear supervised or unsupervised machine learning algorithms [29]. The underlying idea is to apply a kernel function $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ to any pair of training vectors so that the result can be interpreted as an inner product of a mapping function $\phi(\mathbf{x})$, where $\phi : \mathcal{X} \rightarrow F$, and F is a high-dimensional reproducing kernel Hilbert space (RKHS) (a.k.a. the feature space) [30]: $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.

It should be noted that the nonlinear feature mapping $\phi(\cdot)$ is usually unknown. Thus, by means of the kernel function, inner products in the feature space are computed implicitly, i.e. without using the feature vectors directly. This appealing property of kernel methods has then been referred to as the *kernel trick*. If the kernel is a *Mercer kernel*, then for any finite set of samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the matrix whose entries are $k_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite (PD) [18]. This means the resulting kernel matrix has only non-negative eigenvalues, a property that guarantees a convex optimization problem and hence, the existence of a unique solution.

From the exposed, following the recent tendency of kernelizing standard machine learning algorithms, kernel versions of the SOM have been proposed [31], [32], [33], [23], [34], [22], [21], [35]. The kernel SOM variant introduced in [21] and [22], which has been named the gradient descent based KSOM (GD-KSOM) in [23], uses Eq. (1) for finding the winning neuron for a given input pattern. However, its weight updating rule is different from that of the original SOM, being now written as

$$\mathbf{w}_i\,(n+1) = \mathbf{w}_i\,(n) + \eta\,(n)\,h\,(i^*, i, n)\,\nabla J_i(\mathbf{x}(n)), \qquad (5)$$

where firstly we define the function $J_i\,(\mathbf{x}(n))$ as

$$\begin{aligned} J_i(\mathbf{x}(n)) &= \|\phi(\mathbf{x}(n)) - \phi(\mathbf{w}_i(n))\|^2, \qquad (6)\\ &= (\phi(\mathbf{x}(n)) - \phi(\mathbf{w}_i(n)))^T(\phi(\mathbf{x}(n)) - \phi(\mathbf{w}_i(n))),\\ &= \phi(\mathbf{x}(n))^T\phi(\mathbf{x}(n)) + \phi(\mathbf{w}_i(n))^T\phi(\mathbf{w}_i(n))\\ &\qquad -2\phi(\mathbf{x}(n))^T\phi(\mathbf{w}_i(n)),\\ &= k(\mathbf{x}(n),\mathbf{x}(n)) + k(\mathbf{w}_i(n),\mathbf{w}_i(n))\\ &\qquad -2k(\mathbf{x}(n),\mathbf{w}_i(n)), \end{aligned}$$

and then we get the gradient descent vector as follows:

$$\begin{aligned} \nabla J_i(\mathbf{x}(n))) &= \frac{\partial J_i(\mathbf{x}(n))}{\partial \mathbf{w}_i(n)}, \qquad (7)\\ &= \frac{\partial k(\mathbf{w}_i(n),\mathbf{w}_i(n))}{\partial \mathbf{w}_i(n)} - 2\frac{\partial k(\mathbf{x}(n),\mathbf{w}_i(n))}{\partial \mathbf{w}_i(n)}. \end{aligned}$$

An alternative kernel SOM variant was introduced in [23] and has been named as the energy function based Kernel SOM (EF-KSOM). This time the winning neuron is found by the following search procedure:

$$\begin{aligned} i^*(n) &= \arg\min_{\forall i} \|\phi\,(\mathbf{x}(n)) - \phi\,(\mathbf{w}_i(n))\|^2, \qquad (8)\\ &= \arg\min_{\forall i} J_i(\mathbf{x}(n)), \end{aligned}$$

where $J_i(\mathbf{x}(n))$ is already defined in Eq. (6). For classification purposes, the expression $J_i(\mathbf{x}(n))$ can be replaced by

$$\bar{J}_i(\mathbf{x}(n)) = k(\mathbf{w}_i(n),\mathbf{w}_i(n)) - 2k(\mathbf{x}(n),\mathbf{w}_i(n)), \qquad (9)$$

since the term $k(\mathbf{x}(n),\mathbf{x}(n))$ is independent of the neuron index i. Finally, the weight update rule for the EF-KSOM is then given by

$$\mathbf{w}_i\,(n+1) = \mathbf{w}_i\,(n) + \eta\,(n)\,h\,(i^*, i, n)\,\nabla J_i(\mathbf{x}(n)), \qquad (10)$$

where the gradient vector $\nabla J_i(\mathbf{x}(n))$ is defined as in Eq. (7).

It should be noted that the only difference between the GD-KSOM and EF-KSOM is in the way these algorithms select the winning neuron in the feature space. The former executes this operation in the input space, while the latter executes it in the feature space. If a linear kernel function is chosen, i.e. $k(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$, both the GD-KSOM and the EF-KSOM reduces to the original SOM algorithm.

In what concern the choice of the kernel function, many options are available elsewhere. In this paper, we use the Gaussian, the Cauchy and the Log kernel functions, which are described next.

A. Gaussian Kernel function

For two given vectors, $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^p$, the Gaussian kernel function has the following general form:

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\gamma^2}\right), \qquad (11)$$

where $\gamma > 0$ is a scale parameter (a.k.a. the width parameter, in the current context). When this kernel function is inserted into Eq. (6), we get

$$J_i(\mathbf{x}(n)) = 2 - 2\exp\left(-\frac{\|\mathbf{x}(n)-\mathbf{w}_i(n)\|^2}{2\gamma^2}\right), \qquad (12)$$

from which we can obtain the gradient term to be used in the KSOM learning rule:

$$\nabla J_i(\mathbf{x}(n)) = \frac{1}{2\gamma^2}\exp\left(-\frac{\|\mathbf{x}(n)-\mathbf{w}_i(n)\|^2}{2\gamma^2}\right)(\mathbf{x}(n){-}\mathbf{w}_i(n)). \qquad (13)$$

A suitable value for the hyperparameter γ should be carefully tuned to the problem at hand [36]. If it is overestimated, the exponential behaves almost linearly and the projection to high-dimensional feature space loses its nonlinear character. If it is underestimated, the function will lack regularization and decision boundaries tend to become highly sensitive to noise in training data.

B. Cauchy Kernel function

The Cauchy kernel function has the following general form:

$$k(\mathbf{x},\mathbf{y}) = \left(1 + \frac{\|\mathbf{x}-\mathbf{y}\|^2}{\gamma^2}\right)^{-1}, \qquad (14)$$

where $\gamma > 0$ is a scale parameter. Inserting this function into Eq. (6), we get

$$J_i(\mathbf{x}(n)) = \frac{2\gamma^2}{\gamma^2+1} - \frac{2\gamma^2}{\gamma^2 + \|\mathbf{w}_i(n)-\mathbf{x}(n)\|^2}. \qquad (15)$$

The corresponding gradient term for this function is then given by

$$\nabla J_i(\mathbf{x}(n)) = \frac{2\gamma^2(\mathbf{w}_i(n)-\mathbf{x}(n))}{\left(\gamma^2 + \|\mathbf{w}_i(n)-\mathbf{x}(n)\|^2\right)^2}. \qquad (16)$$

This kernel function is a long-tailed kernel, a term borrowed from Probability for denoting distributions in which too small or too large values have large probability to occur, in contrast to the Gaussian distribution for which values far from the mean rarely occur. For this reason, the Cauchy kernel can be used to give long-range influence and sensitivity over the high-dimensional feature space [36].

C. Log Kernel function

This kernel function was introduced in [37] and its expression is given by

$$k(\mathbf{x},\mathbf{y}) = -\log\left(\|\mathbf{x}-\mathbf{y}\|^2 + 1\right), \qquad (17)$$

where $\log$ denotes the natural logarithm. Inserting this function into Eq. (6), we get

$$J_i(\mathbf{x}(n)) = -2\log\left(1 + \frac{\|\mathbf{x}(n)-\mathbf{w}_i(n)\|^2}{\gamma^2}\right), \qquad (18)$$

while the corresponding gradient term for this function is given by

$$\nabla J_i(\mathbf{x}(n)) = \frac{4(\mathbf{x}(n) - \mathbf{w}_i(n))}{\sigma^2 + \|\mathbf{x}(n) - \mathbf{w}_i(n)\|^2}. \qquad (19)$$

The Cauchy kernel function belongs to a class of "not strictly positive definite" kernel functions, named *conditionally definite positive* kernel functions[1], which has been shown anyway to perform very well in many practical applications.

IV. Experimental Testbed and Dataset Generation

A 3-phase squirrel-cage induction motor built by WEG[2] industry is used in this study. Its main characteristics are 0.75 kW (power), 220/380 V (nominal voltage), 3.02/1.75 A (nominal current), 79.5% (efficiency), 1720 rpm (nominal rotational speed), Ip/In = 7.2 (peak to nominal current ratio), and 0.82 (power factor). The dataset is generated with this motor operating in different working conditions. The modules of the laboratory scale test bed are shown in figure 1, and are hereafter explained.

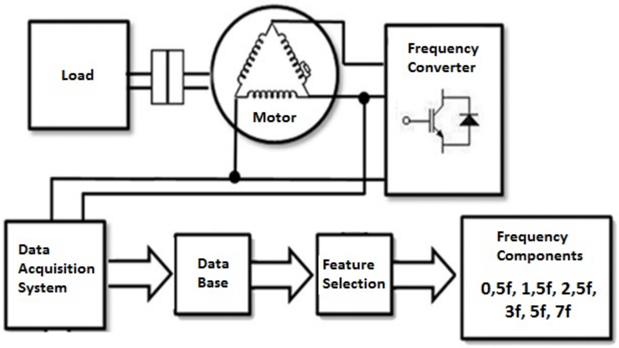

Fig. 1. Modules of the laboratory test bed and the data acquisition system.

Firstly, a Foucault's braking system is used in order to apply three different levels of load: 0% (no load), 50% of nominal load and 100% (full load). Details about the procedure for applying the load are thoroughly described in [12].

In order to vary the speed of the motor, a frequency converter (also known as inverter drive) WEG CFW-09 is utilized with seven different frequencies: 30 Hz, 35 Hz, 40 Hz, 45 Hz, 50 Hz, 55 Hz and 60 Hz. It is worth mentioning that only open loop operation is used with this frequency converter. Moreover, three Hall effect sensors are used to measure the line currents of each phase of this frequency converter.

The motor was rewound so that some extra taps were made available by exposing the stator winding turns[3] of each phase. This was done in order to simulate different inter-turn short-circuit scenarios. In this work, three different levels of fault are used. In the lowest level (level 1), 5 turns were short-circuited, totaling 1.41% of the turns of one phase. In the intermediate level (level 2), 17 turns (4.8%) were short-circuited. Finally, in the highest level (level 3), 32 turns (9.26%) were short-circuited.

TABLE I. Operation Conditions of The Motor

	Components of the feature vector						
Load Level	0%	50%	100%				
Converter Phase	Ph 1	Ph 2	Ph 3				
Converter Frequency	30 Hz	35 Hz	40 Hz	45 Hz	50 Hz	55 Hz	60 Hz
Fault Extension	Normal	HI1	HI2	HI3	LI1	LI2	LI3

An auxiliary command system was built to execute two kinds of short-circuit schemes: high impedance (aiming at simulating the initial low-current state of a short-circuit) and the low impedance. With these two short-circuit schemes and three levels of faults, there are six different fault conditions of the motor. Short-circuit current levels leading to either low or high impedance faults are controlled by resistors in order to protect the motor from permanent damages.

All the operation conditions of the motor are shown in Table I, where the load level applied to the motor, the phase identification, the frequency of the voltage applied by the frequency converter and the *fault extension* are specified. In this last operation condition, the letter **H** denotes a high impedance fault, the letter **L** denotes a low impedance fault, while the numbers 1, 2 and 3 stands for the level of the fault. All these conditions sums up to total of 441 (3 × 3 × 7 × 7) time domain sample vectors.

As shown in Figure 1, the motor was delta connected. In this configuration, two line currents of the frequency converter are directly connected to the faulty phase of the motor. As we aim at developing a monitoring system able to detect faults using just one phase of the converter, just one of these previously mentioned phases was used in order to avoid redundancy of information. Thus, 294 samples are used: 147 from phase 1 (directly connected to the fault current) and 147 from phase 3 (indirectly connected to the fault current). These samples are represented in figure 2.

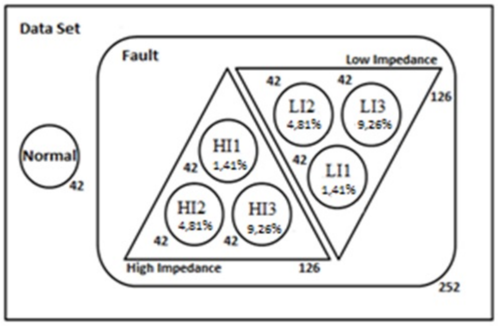

Fig. 2. Data Acquisition Scheme.

As can be inferred from Table I, the task of interest can be approached as a multi-class problem, if one considers each fault extension as a class (normal, H1, H2, H3, L1, L2, L3). As such, each class has 42 samples. Alternatively, one can rearrange data samples into three classes, namely: normal condition (with 42 samples), high impedance fault (with 126 samples, merging the classes HI1, HI2 and HI3)

[1] Let $\mathcal{X}$ be a nonempty set. A kernel $k(\cdot,\cdot)$ is called *conditionally positive definite* if and only if it is symmetric and $\sum_{j,k}^{n} c_j c_k k(\mathbf{x}_j, \mathbf{x}_k) \geq 0$, for $n \geq 1$, $c_1, \ldots, c_n \in \mathbb{R}$ with $\sum_{j=1}^{n} c_j = 0$ and $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$.

[2] http://www.weg.net/institutional/BR/en/

[3]

or low impedance fault (also with 126 samples, merging the classes LI1, LI2 and LI3).

In this work, however, we are interested in detecting any kind of fault, independently of the impedance level. Hence, the problem is treated henceforth as a binary classification problem: normal (class 1) or faulty condition (class 2). By doing so, there are 252 samples in class 2, resulting from the merging of the classes H1, H2, H3, L1, L2 and L3. Class 1 remains with 42 samples.

In this dataset, by “sample” we mean a current signal stored as a vector of 100,000 components, resulting from 10 seconds of acquisition with a 10 kHz sampling frequency. To generate the feature vectors for classification purposes, the Fast Fourier Transform (FFT) is used. The procedure for building the feature vector for each current signal is comprised of the following steps:

Step 1 - Define the load condition of the motor.

Step 2 - Define the fundamental frequency (f_c) of the converter drive.

Step 3 - Read the current signal for 10s at a 10KHz sampling rate.

Step 4 - Apply the FFT to the current signal. Since the output of the FFT is comprised of a sequence of complex numbers, take their absolute values.

Step 5 - Find the frequency corresponding to the maximum value of the computed spectrum. Denote it as $\hat{f}_c$, since it is an estimate of f_c (see Step 2).

Step 6 - Build the associated 6-dimensional feature vector by selecting the corresponding FFT output values for the following harmonics of $\hat{f}_c$: $\{0.5\hat{f}_c,\ 1.5\hat{f}_c,\ 2.5\hat{f}_c,\ 3\hat{f}_c,\ 5\hat{f}_c,\ 7\hat{f}_c\}$.

The rationale for using the *current signature* approach is given in [38], who reports that there are no novel components in stator motor current frequency spectrum due to the short-circuit fault, but rather an increase in some existing components. In [12], the estimated converter’s fundamental frequency ($\hat{f}_c$) itself and the second harmonic ($2\hat{f}_c$) were also used as input features for a SOM-based classifiers in addition to the aforementioned six ones (see Step 6). However, in [39] those two additional features were not found to be relevant for fault detection purposes.

In summary, the dataset is comprised of 294 6-dimensional feature vectors (252 labelled as faulty, 42 labelled as normal), in which the attribute values represents the FFT values for the chosen 6 harmonics of the fundamental frequency of the converter drive. In the next section we present the results of performance comparison carried out in this work.

V. Results and Discussion

For all the following experiments, the feature vectors are normalized to zero mean and unit variance. All neural models were implemented from scratch using the MATLAB software, version R2013a (8.1.0.604), running on Windows 10 Home, installed in a Dell notebook, Core i7, 1.60GHz, 8GB RAM.

In the following experiments, seven KSOM variants are evaluated: the standard two-dimensional SOM (SOM-2D); the KSOM-GD with three different kernels - Gaussian (KSOM-GD-G), Cauchy (KSOM-GD-C) and Log (KSOM-GD-L); and the KSOM-EF with the same kernels (KSOM-EF-G, KSOM-EF-C, KSOM-EF-L). The three labeling methods described in Subsection II-A are used to convert the SOM and the KSOM variants into pattern classifiers. They are identified by letters as follows: minimum distance (MD), average distance (AD) and majority voting (MV). After intensive prior experimentation, the hyperparameter γ for the three kernel functions was set to $\gamma = 0.5$. A rectangular grid of dimensions 5×4 is used for all the SOM and KSOM classifiers evaluated in this paper.

TABLE II. ACCURACIES OF THE EVALUATED SOM AND KSOM CLASSIFIERS FOR DIFFERENT LABELING METHODS.

Labeling method	**MD**		**AD**		**MV**	
Classifier	**max**	**mean**	**max**	**mean**	**max**	**mean**
SOM-2D	0.966	0.829	0.949	0.858	0.949	0.736
KSOM-GD-G	0.949	0.829	0.932	0.858	0.932	**0.773**
KSOM-GD-L	0.915	0.831	0.949	**0.863**	0.898	0.758
KSOM-GD-C	0.949	0.826	0.949	0.854	0.915	0.756
KSOM-EF-G	0.932	0.827	0.949	0.858	0.881	0.765
KSOM-EF-L	0.949	**0.861**	0.949	0.858	0.881	0.727
KSOM-EF-C	0.949	**0.861**	0.949	0.844	0.898	0.737

For each evaluated classifier, 50 independent training-labeling-testing runs are executed. For each run, four steps are performed, namely: (i) holdout (partition of the data between training and test sets), (ii) unsupervised training, (iii) neuron labeling and (iv) performance testing. For the holdout step, the data is randomly divided as follows: 80% for training and remaining 20% for test. At the end of test phase, several statistical figures of merit for the performance of each classifier are computed.

The achieved values of the figures of merits are reported in Table II, for all evaluated kernel SOM-based classifiers. Best average performances are highlighted in bold for each labeling method. If we take the best performances for each labeling method in separate, we have the following relevant results: the best performances for the MD method were achieved by far by the KSOM-EF-L and KSOM-EF-C classifiers; while, for the AD method the best performance was achieved by the KSOM-GD-L classifier. For the MV method, the best performance was achieved by the KSOM-GD-G classifier, but this performance is quite inferior to those reported by any other classifier using the MD and AD labeling methods.

By contrasting the results reported in Table II for the MD and AD methods, it is possible to infer that the former leads to mean values with higher variance across the classifiers than the latter. This behavior is more easily observed in Figures 3 and 4, where we show the boxplots of the accuracy rates for all evaluated classifiers. For the sake of completeness, the boxplots of the accuracy rates for all evaluated classifiers using the majority voting method is shown in Figure 3.

In Figure 3 it is evident that the best performances were achieved by the KSOM-EF-L and KSOM-EF-C classifiers when the MD method was used. These classifiers also presented the smaller dispersion in comparison to the others. In Figure 4 the performance of the KSOM-GD-L classifier is closely followed by that of the standard SOM-2D classifier. This occurred because, as mentioned before, the use of the AD method led to a lower variance of the mean values across the classifiers, improving the overall performances of all classifiers as a whole.

In Tables III and IV, we report the confusion matrices for the best performing classifiers using the MD labeling method. In Table V we report the confusion matrices for the best performing classifier using the AD labeling method. In these tables we show the confusion matrices of the classifier realizations that produced the best and worst values for the accuracy rate among the 50 training-testing runs.

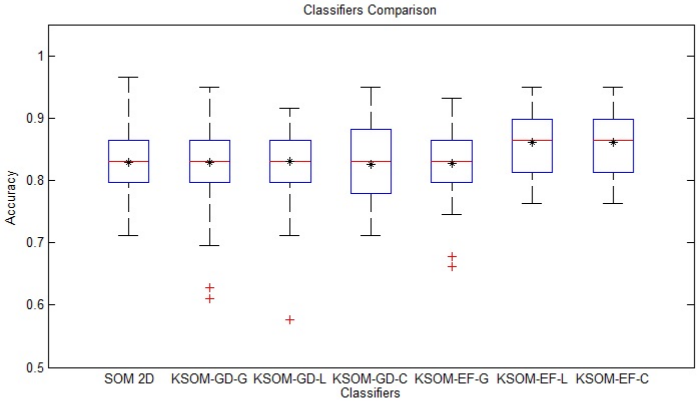

Fig. 3. Boxplots of the achieved accuracy rates for all evaluated classifiers using the minimum distance labeling method.

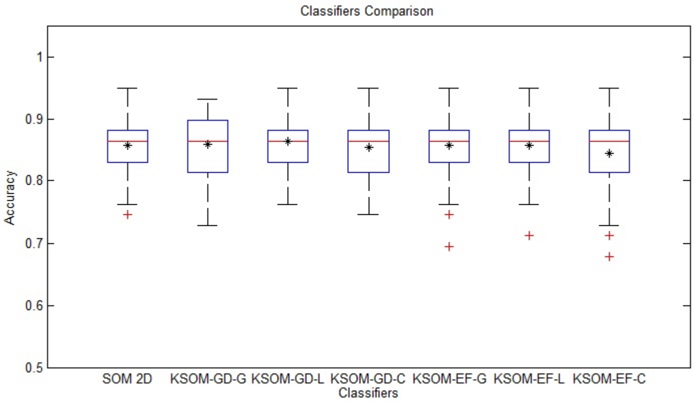

Fig. 4. Boxplots of the achieved accuracy rates for all evaluated classifiers using the average distance labeling method.

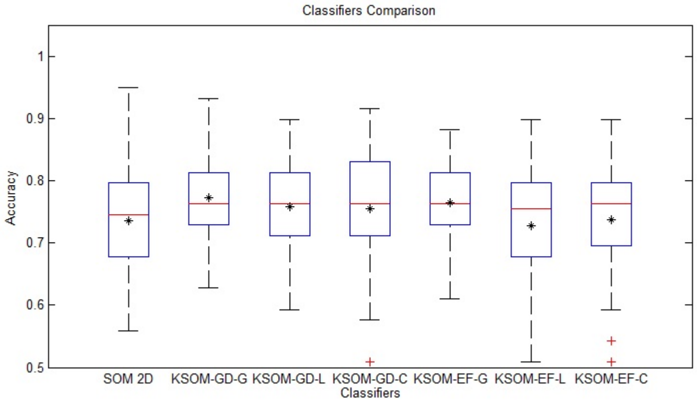

Fig. 5. Boxplots of the achieved accuracy rates for all evaluated classifiers using the majority voting labeling method.

It can be observed in the confusion matrices a tendency to misclassify normal samples as faulty ones (a false positive error). This behavior can be explained by the fact that the dataset is highly unbalanced, with much more faulty examples than normal ones. This problem can be somewhat alleviated by resampling normal samples during training with higher probability than the faulty ones. Inclusion of artificially generated normal samples in the training data can also reduce false positive rates, improving the classification performance as a whole.

TABLE III. CONFUSION MATRICES FOR THE KSOM-EF-L CLASSIFIER USING THE MD LABELING METHOD (BEST AND WORST CASES).

BEST CASE			WORST CASE		
KSOM-EF-L	**Actual Class**		**KSOM-EF-L**	**Actual Class**	
Predicted class	**Normal**	**Faulty**	**Predicted class**	**Normal**	**Faulty**
Normal	2	0	**Normal**	7	0
Faulty	1	56	**Faulty**	7	45

TABLE IV. CONFUSION MATRICES FOR THE KSOM-EF-C CLASSIFIER USING THE MD LABELING METHOD (BEST AND WORST CASES).

BEST CASE			WORST CASE		
KSOM-EF-C	**Actual Class**		**KSOM-EF-C**	**Actual Class**	
Predicted class	**Normal**	**Faulty**	**Predicted class**	**Normal**	**Faulty**
Normal	2	0	**Normal**	8	0
Faulty	1	56	**Faulty**	6	45

VI. CONCLUSIONS AND FURTHER WORK

In this paper, we reported the results of a comprehensive application of kernelized variants of the SOM algorithm for the detection of short circuit faults in converter-fed three-phase induction motors. We have shown that the obtained results are highly dependent on the neuron labeling method and also on the kernel function. As could be inferred from the presented results, the average distance labeling method seemed to be more stable, leading to a lower variance of the mean accuracy values across the classifiers, improving the overall performances of all classifiers as a whole, including the one based on the standard SOM. Among the three kernel functions used in the experiments, the *log kernel function* led to the best performing classifier.

Currently, we are evaluating the KSOM-based classifiers described in this paper on the same fault detection problem, but this time we are following the *novelty/anomaly detection* paradigm [40], [41]. According to this paradigm, only one of the classes (usually, the one with higher number of examples) is modeled/learned by a single classifier. New incoming samples which do not fit to the learned model are classified as anomalous. Based on the first results, this approach is very promising.

ACKNOWLEDGMENTS

The authors thank CNPq for the financial support (grant no. 309451/2015-9) and the Federal Institute of Ceará (IFCE) for providing the physical structure and testbed used to generate the data set used in this work.

TABLE V. CONFUSION MATRICES FOR THE KSOM-GD-L CLASSIFIER USING THE AD LABELING METHOD (BEST AND WORST CASES).

BEST CASE			WORST CASE		
KSOM-GD-L	**Actual Class**		**KSOM-GD-L**	**Actual Class**	
Predicted class	**Normal**	**Faulty**	**Predicted class**	**Normal**	**Faulty**
Normal	1	0	**Normal**	1	0
Faulty	2	56	**Faulty**	6	52

References

[1] V. N. Ghate and S. V. Dudul, “Optimal MLP neural network classifier for fault detection of three phase induction motor,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 3468–3481, 2010.

[2] W. T. Thomson and M. Fenger, “Current signature analysis to detect induction motor faults,” *IEEE Industry Applications Magazine*, vol. 7, no. 4, pp. 26–34, 2001.

[3] K. Bacha, H. Henao, M. Gossa, and G.-A. Capolino, “Induction machine fault detection using stray flux EMF measurement and neural network-based decision,” *Electric Power Systems Research*, vol. 78, no. 7, pp. 1247–1255, 2008.

[4] S. Nandi, H. A. Toliyat, and X. Li, “Condition monitoring and fault diagnosis of electrical motors-a review,” *IEEE Transactions on Energy Conversion*, vol. 20, no. 4, pp. 719–729, 2005.

[5] J. Vico and R. Hunt, “Protection principles for electrical motors in the cement industry,” in *Proceedings of the 2010 IEEE-IAS/PCA 52nd Cement Industry Technical Conference*, 2010, pp. 1–13.

[6] A. H. Bonnett, “Root cause failure analysis for ac induction motors in the petroleum and chemical industry,” in *2010 Record of Conference Papers Industry Applications Society 57th Annual Petroleum and Chemical Industry Conference (PCIC)*, 2010.

[7] J. F. Martins, V. F. Pires, and A. Pires, “Unsupervised neural-network-based algorithm for an on-line diagnosis of three-phase induction motor stator fault,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 259–264, 2007.

[8] R. Natarajan, “Failure identification of induction motors by sensing unbalanced stator currents,” *IEEE Transactions on Energy Conversion*, vol. 4, no. 4, pp. 585–590, 1989.

[9] R. M. Tallam, S.-B. Lee, G. Stone, G. B. Kliman, J. Yoo, T. G. Habetler, and R. G. Harley, “A survey of methods for detection of stator related faults in induction machines,” in *Proceedings of the 4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED’2003*, 2003, pp. 35–46.

[10] W. T. Thomson, “On-line MCSA to diagnose shorted turns in low voltage stator windings of 3-phase induction motors prior to failure,” in *Proceedings of the IEEE International Electric Machines and Drives Conference (IEMDC)’2001*, 2001, pp. 891–898.

[11] D. Asfani, A. Muhammad, M. Purnomo, T. Hiyama *et al.*, “Temporary short circuit detection in induction motor winding using combination of wavelet transform and neural network,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 5367–5375, 2012.

[12] D. N. Coelho and C. M. S. Medeiros, “Short circuit incipient fault detection and supervision in a three-phase induction motor with a SOM-based algorithm,” in *Advances in Self-Organizing Maps*. Springer, 2013, pp. 315–323.

[13] T. Kohonen, “Essentials of the self-organizing map,” *Neural Networks*, vol. 37, pp. 52–65, 2013.

[14] J. Seshadrinath, B. Singh, and B. K. Panigrahi, “Incipient interturn fault diagnosis in induction machines using an analytic wavelet-based optimized bayesian inference,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 990–1001, 2014.

[15] D. N. Coelho, G. A. Barreto, C. M. S. Medeiros, and J. D. A. Santos, “Performance comparison of classifiers in the detection of short circuit incipient fault in a three-phase induction motor,” in *Proceeding of the 2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES’2014)*, 2014, pp. 42–48.

[16] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Pearson, 2008.

[17] G. B. Huang, Q. Y. Zhu, and C. K. Ziew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[18] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.

[19] J. A. K. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[20] A. H. Souza Jr., F. Corona, G. A. Barreto, Y. Miche, and A. Lendasse, “Minimal learning machine: A novel supervised distance-based approach for regression and classification,” *Neurocomputing*, vol. 164, pp. 34–44, 2015.

[21] P. Andras, “Kernel-Kohonen networks,” *International Journal of Neural Systems*, vol. 12, no. 02, pp. 117–135, 2002.

[22] Z. S. Pan, S. C. Chen, and D. Q. Zhang, “A kernel-based SOM classification in input space,” *Acta Electronica Sinica*, vol. 32, no. 2, pp. 227–231, 2004.

[23] K. W. Lau, H. Yin, and S. Hubbard, “Kernel self-organising maps for classification,” *Neurocomputing*, vol. 69, no. 16, pp. 2033–2040, 2006.

[24] M. van Hulle, “Self-organizing maps,” in *Handbook of Natural Computing: Theory, Experiments, and Applications*, G. Rozenberg, T. Baeck, and J. Kok, Eds. Springer-Verlag, 2010, pp. 1–45.

[25] H. Yin, “The self-organizing maps: Background, theories, extensions and applications,” in *Computational Intelligence: A Compendium*, ser. Studies in Computational Intelligence, J. Fulcher and L. C. Jain, Eds. Springer-Verlag, 2008, vol. 115, pp. 715–762.

[26] T. K. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, “Engineering applications of the self-organizing map,” *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1358–1384, 1996.

[27] M. Biehl, B. Hammer, and T. Villmann, “Prototype-based models in machine learning,” *WIREs Cognitive Science*, vol. 7, no. 2, pp. 92–111, 2016.

[28] C. L. C. Mattos and G. A. Barreto, “ARTIE and MUSCLE models: building ensemble classifiers from fuzzy ART and SOM networks,” *Neural Computing and Applications*, vol. 22, no. 1, pp. 49–61, 2013.

[29] F. Jäkel, B. Schölkopf, and F. A. Wichmann, “A tutorial on kernel methods for categorization,” *Journal of Mathematical Psychology*, vol. 51, no. 6, pp. 343–358, 2007.

[30] H. Yin, “On the equivalence between kernel self-organising maps and self-organising mixture density networks,” *Neural Networks*, vol. 19, no. 6, pp. 780–784, 2006.

[31] M. Olteanu, N. Villa-Vialaneix, and C. Cierco-Ayrolles, “Multiple kernel self-organizing maps,” in *Proceedings of the 21st European Symposium on Artificial Neural Networks (ESANN’2013)*, 2013, pp. 83–88.

[32] N. Chen and H. Zhang, “Extended kernel self-organizing map clustering algorithm,” in *Proceedings of the 5th International Conference on Natural Computation (ICNC’2009*, 2009, pp. 454–458.

[33] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, “A survey of kernel and spectral methods for clustering,” *Pattern Recognition*, vol. 41, pp. 176–190, 2008.

[34] R. Inokuchi and S. Miyamoto, “LVQ clustering and SOM using a kernel function,” in *Proceedings of 2004 IEEE International Conference on Fuzzy Systems*, vol. 3, 2004, pp. 1497–1500.

[35] D. Macdonald and C. Fyfe, “The kernel self-organising map,” in *Proceedings of 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES’2000)*, vol. 1, 2000, pp. 317–320.

[36] C. R. de Souza, “Kernel functions for machine learning applications.” [Online]. Available: http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/

[37] S. Boughorbel, J.-P. Tarel, and N. Boujemaa, “Conditionally positive definite kernels for SVM based image recognition,” in *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME’2005)*, 2005, pp. 1–4.

[38] G. Joksimovic and J. Penman, “The detection of inter-turn short circuits in the stator windings of operating motors,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 1078–1084, 2000.

[39] A. G. Oliveira and C. M. S. Medeiros, “Stator winding interturns short circuit fault detection in a three phase induction motor driven by frequency converter using neural networks,” in *Energy Efficiency in Motor Driven Systems 2013 Conference, EEMODS*, 2013.

[40] S. Ma, S. M. Li, and Y. P. Xiong, “Uncertainty reduced novelty detection approach applied to rotating machinery for condition monitoring,” *Shock and Vibration*, vol. 2015, no. ID 737213, pp. 1–10, 2015.

[41] G. A. Barreto and R. A. Frota, “A unifying methodology for the evaluation of neural network models on novelty detection tasks,” *Pattern Analysis and Applications*, vol. 16, no. 1, pp. 83–97, 2013.