# Neural Network used to Stator Winding Interturn Short-Circuit Fault Detection in an Induction Motor Driven By Frequency Converter

*Átila Girão de Oliveira; Ricardo Silva Thé Pontes*
Electric Engineering Departament
Universidade Federal do Ceará, UFC
Fortaleza, Brasil
atilagirao@yahoo.com.br; ricthe@dee.ufc.br

*Cláudio Marques de Sá Medeiros*
Industrial Departament
Instituto Federal de Educação, Ciência e Tecnologia do Ceará, IFCE
Fortaleza, Brasil
claudiosa@ifce.edu.br

***Abstract*—This work is the application of a Multilayer Perceptron Artificial Neural Network (MLP ANN) to detect early interturn short-circuit faults in a three-phase converter-fed induction motor. The quantity used to analyze the problem is the stator current or, more specifically, the harmonic content of its frequency spectrum, also called current signature. The analysis through the current signature is a non-invasive method and may be embedded in the frequency converter, what is a great advantage. The dataset used for training and validating the ANN is obtained using a test bench that allows applying different levels of interturn short-circuits in the machine. It is observed that the fault motor dataset and healthy motor dataset are difficult to separate, which demands a large computational effort to choose a proper MLP topology. The MLP is trained by two different algorithms (the classical error Backpropagation - BP - and an adaptation of the newer Extreme Learning Machine - ELM) and the results are thoroughly explored, including after the application of a pruning method called CAPE. Then it is slightly compared with the results of a Self-Organized Map ANN [1] obtained by using the same dataset.**

***Keywords— Backpropagation; Extreme Learning Machine; Fault Detection; Frequency Converter Multi-layer Perceptron; Stator Winding Interturn short-circuit.***

## I. INTRODUCTION

The induction machine is consolidated as main motor force in industries. According Thomson and Fenger [2], in an industrialized nation, the induction motor may consume typically among 40% and 50% of all capacity generated.

Despite the recognized robustness and reliability of this machine, it is subject to fault occurrence. The more common occurrences are bearing faults, stator or rotor isolation faults, open bars or crack of the rings and eccentricity fault [3].

In industries the cost of an unscheduled downtime is too high; hence they invest increasingly to improve their maintenance programs. Faults like open rotor bars, eccentricity and bearing faults take time to evolve and to put the motor out of operation. In this context, the constant online monitoring is important to early fault detection in a way to have time to program a maintenance order.

The stator winding interturn short-circuit (SWITSC) takes a short time to evolve and to condemn the motor. Thomson and Fenger [2] tested low voltage three-phase induction machines from early SWITSC until complete failure and found that exist a time of a few minutes to occur the fault evolution. In this case the early detection makes possible to repair the machine by rewinding it or, in large by machines, removing short-circuited coils; and the early operation stopping avoids electrical arcs due short-circuit which offers an additional protection to areas where there are explosion risks. Moreover, after the severe fault, the machine ferromagnetic core is damaged and it becomes probably irreparable.

It has become more and more common the use of frequency converter to drive induction motors. That gives more versatility to the drive machines because it allows application with varying rotation speed [4]. The high current in the motor due to fault also affects the frequency converter integrity, so an early detection means an additional protection. Moreover, once the electronics is already in use, could be advantageous have detection systems previously embedded in it. In past years the developing of several computational intelligence techniques has added new possibilities to fault detection and diagnosis systems, most of them potentially suitable to be embedded in electronic converters.

The most of researches uses line-fed machines. Using converter-fed motors, Kowalski and Wolkiewicz [5] analyze the spectrum of instantaneous Park power and torque signals to diagnosis early SWITSC and broken rotor bars. Coelho and Medeiros [1] try to map the SWITSC fault using self-organized map and then classify the motor.

According Nandi et al [3], isolation fault represents from 30% to 40% of all kinds of faults reported in induction motors. In this work is investigated the motor current signature analysis (MCSA) together with the potential of a single hidden-layer feed-forward neural network (SLFN) classifier as a tool of SWITSC detection to converter-fed three-phase induction machine. Two algorithms were used to training the neural networks: the error backpropagation (BP), and the ELM. The datasets were obtained by an experimental test bench where different levels of SWITSC can be applied.

The section II shows the problem of the SWITSC and how it has been treated past years. Section III covers the basic

978-1-4799-3194-1/13 $31.00 © 2013 IEEE
DOI 10.1109/BRICS-CCI-CBIC.2013.82

about the neural algorithms used in the paper. In section IV the laboratory tests and data acquisition are explained as well as the ANN attributes selection. The classification results are in section V and the section VI concludes this work.

## II. Stator Winding Interturn Short-Circuit Overview

Isolation systems are submitted to several kinds of efforts that might cause a failure. Due to the use of inverters to drive electrical motors with a frequency switching typically of about 10 kHz, occur voltage peaks such that increases considerably the machine isolation stress. As result, the stress of converter-fed motor might be until ten times higher than line fed machines [6].

The failure process is usually initialized as a turn-to-turn high impedance fault (order of kΩ) in the same phase, between phases or phase-to-ground [7]. The fault current can reaches two times the rotor blocked current, which causes high localized heating and makes the fault quickly spreads.

Different methods have been used in many researches to detect stator interturn short-circuit. Considering the MCSA method, Joksimovic and Penman [8] show that there are no novel components in the spectrum of stator motor current frequency due to the isolation fault. In fact it was observed that only occurs an increase of the existents components. Stavrou et al [9] search in the current frequency spectrum for the variation in frequencies as function of number of poles, slots and slip, that is, specific constructive features.

Penman et al [10] develop an equation (1) to calculate which harmonic components in the axial leakage flux waveform are functions of the SWITSC and then they propose a method to detect the fault by monitoring those components.

$$f_{st} = \{k \pm n(1-s)/p\} f_1 \quad (1)$$

The $f_{st}$ are components in function of the interturn short-circuit, $k = 1, 3, 5...$, means the temporal harmonics order, $n = 1, 2, 3...$, means the spatial harmonic order, $s$ is the slip, $p$ is the pair of poles, $f_1$ is the power supply fundamental frequency. Several kinds of faults affect the current spectrum and some harmonics are affected by more than one abnormal condition, so one must be careful to choose the correct frequencies that will be used to indicate the problem.

Thomson and Fenger [2] expand the concept of the leakage flux to stator currents, once the flux inside the machine also crosses the stator windings. They do an experimental analysis in low voltage motors to verify which the frequencies that are function only of the short-circuit and no other conditions as unbalanced phases, misalignment of the shaft, broken rotor bars, bearing faults, etc. The components found in [2] using (1) as function only of short-circuit are $f_{st1}$ when $k=1, n=3$ and $k=1, n=5$; to an unload motor ($s \approx 0$) with 2 pair of poles, these harmonics would be $2.5f_1$ and $3.5f_1$.

Also using MCSA, Gazzana et al [11] create a system to early detection and diagnosis rotor broken bars, air-gap eccentricities and SWITSC in induction motors. To SWITSC (1) is used with k=1, n=7 and the Welch's method is used to obtain the frequency spectrum. The choice of a high order spatial component is due to the low order eccentricity components are coincident with the short-circuit components.

Bouzid et al [12] use ANN to locate the phase where the short-circuit is. It is chosen as fault feature the phase shift between the three phase voltages and currents. The detection is made by a MLP with 3 outputs, each one referent to one phase. If that neuron was active it means a fault in that phase.

Das et al [13] process the line current signal recorded from motor terminals through a Park's transformation followed by Continuous Wavelet Transformation and they use a Support Vector Machine (SVM) to classify the extracted features. From the 18 test cases used for prediction, a total of 16 fault cases were correctly identified by the proper SVM.

Among all possible methods to fault detection MCSA has a great potential because it is non-invasive; does not require installation of sensor in the machine; does not require adaptation to classified areas; presents high capacity of remote monitoring; can be applied to any machine, with no power restriction; presents sensitivity to mechanical faults, stator electric faults and feed problems [14]. To these advantages may be added, to converter-fed motors, the possibility of to embed the detection system in the own converter, especially if a computational intelligence technique was used.

This work presents MLP ANNs to classification of SWITSC. The equation (1) developed by the Penman's theory and expanded by Thompson is initially used for feature extraction to the fault detection.

## III. Neural Networks in a Nutshell

It was proved that an MLP ANN with one hidden layer can approximate any continuous function with a certain precision since it has neurons enough [15]. In classification they are recommended for applications in which there is an unknown non-linear relation between input and output dataset, even for complex non-linear multivariable problems. They are capable to learn this relationship by data presentation, and then generalize the knowledge and classify new data.

In Fig. 1 is shown a generic architecture of a SLFN. First there is the input layer that is fully connected with the hidden layer by the weights $w_{ij}$ just like the synapses that connect biological neurons. The hidden layer uses non-linear function to make a transformation in the data space and produces a linear-separable data space. The hidden-layer output is the input to the output layer, where the classification is done.

The MLP trained by BP algorithm (MLP/BP) is probably the most studied and classical neural model, especially in classification applications, but even nowadays, a user soon becomes aware of the difficulties in finding an optimal architecture for real-world applications.

The many parameters that need to be adjusted by heuristic rules or more commonly by attempt and error demands much time and effort to design a proper MLP, also the time wasted with the BP training is usually elevated. That led many researchers to look for a better algorithm.

A new learning algorithm for SLFNs named ELM was presented in [16] and has become the aim of many studies. The

two great advantages of the ELM are the easy neural network design, which has practically no parameters to adjusting and the training algorithm that computes extremely fast.

*A. Backpropagation*

This section describes briefly the most common algorithm used to training the MLP ANN, a detailed version can be found in [17]. The learning algorithm requires two passes of computation: a forward pass and a backward pass. The forward pass leads to the output given by

$$y_k^{(o)}(t) = \varphi_k\left[\sum_{i=0}^{Q} m_{ki}\varphi_i\left(\sum_{j=0}^{P} w_{ij}x_j\,(t)\right)\right], \quad (1)$$

where $w_{ij}$ is the weight connecting the input $_j$ to the hidden neuron $i$, $Q$ is the number of hidden neurons, $P$ is the dimension of the input vector, $\varphi(\cdot)$ are the activation functions, $m_{ki}$ is the weight connecting the hidden neuron $i$ to the output neuron $k$ *(k=1,…,M)*, and $M$ is the number of output neurons. In the backward pass, first is computed the error value $e_k^{(o)}(t)$ generated by each output neuron at time step $t$

$$e_k^{(o)}(t) = d_k(t) - y_k^{(o)}(t), k = 1, \dots, M \quad (2)$$

where $d_k(t)$ is the target output value for the output neuron $k$. The error signal $e_k(t)$ should be multiplied by the derivative $\phi_k'\left[u_k^{(o)}(t)\right] = \partial\phi_k/\partial u_k^{(o)}$ before being back-propagated, thus generating the so called *local gradient* of the output neuron $k$

$$\delta_k^{(o)}(t) = \phi_k'\left[u_k^{(o)}(t)\right]e_k^{(o)}(t) \quad (3)$$

Similarly, the local gradient $\delta_i^{(h)}(t)$ of the hidden neuron $i$ is then computed as

$$\delta_i^{(h)}(t) = \phi_i'\left[u_i^{(h)}(t)\right]\sum_{k=1}^{M} m_{ki}(t)\delta_k^{(o)}(t) = \phi_i'\left[u_i^{(h)}(t)\right]e_i^{(h)}(t), \quad i = 0, \dots, Q \quad (4)$$

Where the term $e_i^{(h)}(t)$ is the *back-propagated* error signal for the hidden neuron $i$. Finally, the synaptic weights of the neurons are updated according to the following rules

$$m_{ki}(t+1) = m_{ki}(t) + \eta\delta_k^{(o)}(t)y_i^{(h)}(t), i = 0, \dots, Q, \quad (5)$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_i^{(h)}(t)x_j(t), \quad j = 0, \dots, P, \quad (6)$$

where $\eta$ is the learning rate.

One complete presentation of the entire training set is called an epoch. It is good practice to randomize the order of presentation of training examples from one epoch to the next, in order to make the search in the weight space stochastic over the learning cycles. A simple way of evaluating convergence is through the average squared error

$$\varepsilon_{train} = \frac{1}{2N}\sum_{t=1}^{N}\sum_{k=1}^{M}\left[d_k(t) - y_k^{(o)}(t)\right]^2, \quad (7)$$

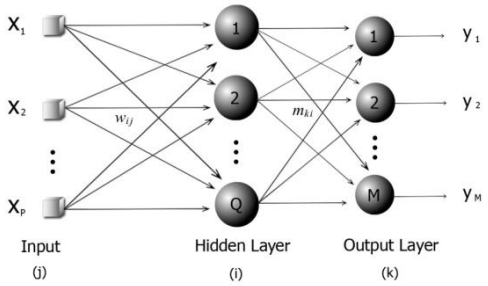


Figure 1. Generic Single-Hidden Layer Feedforward Neural Network.

computed at the end of training run using the training data vectors. If it falls below a pre-specified value then convergence is achieved. The generalization performance of the MLP should be evaluated on a testing set, which contains examples not seen before by the network.

*B. Extreme Learning Machine (ELM)*

The ELM algorithm was proposed in 2004 by Guang-Bin et al [15] as an attractive option to be used to training SLFNs instead of classical gradient descent-based methods like the BP. The authors prove that the proposed algorithm typically can train any dataset thousands of times faster than the error BP. It follows a possibility to the ELM algorithm:

The hidden layer input and output can be expressed in a matrix-vector notation respectively as (8) and (9),

$$u(t) = Wx(t), \quad (8)$$

$$y^{(h)}(t) = \varphi_i\big(u_i(t)\big) = \varphi_i\big(Wx(t)\big), \quad (9)$$

where $W$ is an arbitrary matrix of weights that connects the input with the hidden layer. The function $\varphi_i(\cdot)$ is applied to each one of the Q components of vector $u(t)$. The vector $y^{(h)}(t)$ is calculated to each dataset sample and organized in a matrix $Y^{(h)}$ with $Q$ (number of hidden neurons) lines and $N$ (number of samples) rows. This matrix is used to calculate the weights that connect the hidden and the output layer.

To each input vector $x(t), t = 1, \dots, N,$ there exists an output target vector $d(t)$. The $N$ output target vectors can be organized in a matrix with $M$ (number of output neurons) lines and $N$ rows.

$$D = [d(1) \mid d(2) \mid \cdots \mid d(N)]_{MxN} \quad (10)$$

The calculation of the output weights can be considered as a calculation of a linear mapping between the hidden layer and the output layer. That means finding the matrix $M$ that better represents the transformation of the input vectors $x(t)$ to its correspondent target vector $d(t)$,

$$d(t) = My^{(h)}(t) \quad (11)$$

This can be done by the least-square error method, also known as pseudo-inverse method. The expression is given by

$$M = DY^{(h)^T}(Y^{(h)}Y^{(h)^T})^{-1} \quad (12)$$

### C. Self-Organizing Map (SOM)

The SOM is a competitive and unsupervised network that performs a mapping from an input space to a discrete output space, preserving the topological properties of the input [18].

For each input vector randomly chosen sequentially, Euclidean distances are calculated in relation to the neurons of the network. The one that presents less distance will be the winner neuron. The winner and its neighbors have their weights updated according to the competitive rule

$$w_i(t+1) = w_i(t) + \aleph_{i,i^*}(t)\eta(x(t) - w_i(t)), \qquad (13)$$

in which, $\aleph_{i,i^*}$ function is a neighborhood function centered at the winning neuron. The weights move toward the input in a straight line proportional to the size of $\eta$. The SOM mentioned in this paper uses a two-dimensional arrangement of neurons to map the input space into output space.

### D. Cross-Correletion Analysis of back-Propagated Erros (CAPE)

The CAPE algorithm is a weight pruning method based on the following general principle called the *Principle of Maximum Correlation of Errors*: Relevant synaptic weights tend to generate higher correlations between error signals associated with the neurons of a given layer and the error signals propagated back to the previous layer. Prunable weights tend to generate smaller correlations. In Eq. 4, it was defined the backpropagated error signal for the $i$th hidden neuron as,

$$e_i^{(h)}(t) = \sum_{k=1}^{M} m_{ki}(t)\delta_k^{(o)}(t) = \sum_{k=1}^{M} m_{ki}^*(t)e_k^{(o)}(t), \qquad (14)$$

we can note that $e_i^{(h)}(t)$ is the result of a linear combination of the output error signals $e_k^{(o)}(t), k = 1, \dots, M$, with weights $m_{ki}^*(t) = m_{ki}(t)\varphi_k'\left[u_k^{(o)}(t)\right]$. Thus, the higher the value of $m_{ki}^*(t)$ and, hence, of $m_{ki}(t)$, the higher the correlations between $e_i^{(h)}(t)$ and $e_k^{(o)}(t)$. A similar linear relationship can be derived for the signals $e_i^{(h)}(t)$, associated with the hidden neurons, and the signals $e_j^{(i)}(t)$, associated with the input units. Details about the algorithm can be found in [19].

## IV. Experimental Data Aquisition

To collect the dataset needed to training the ANNs, an induction motor was rewound by a specialized company. The motor is a WEG with rated values: 0,75 kW (1.0 CV), 60 Hz, 220/380 V, 3.02/1.75 A, n=1720 rpm, efficiency: 79.5%, F.P. = 0.82. A frequency converter was used to drive the motor and a Foucault's brake was used to apply load. The data was acquired by a data acquisition system with 16 bits of resolution, passing through a 1 kHz analog filter and a signal amplifier; that implies the frequency band used to work is limited to 500 Hz by the Nyquist's[1] theorem.

Three phase current signals are measured by hall sensors and collected with a frequency of sample of 10 kHz during 10 seconds. That creates a dataset of 100,000 samples to each phase. The motor is delta connected. To the acquisition the frequency converter are set to seven different values: 30 Hz, 35 Hz, …, 60 Hz. Furthermore, three load conditions are considered: unload, 50% and 100% of rated load.

To emulating the incipient SWITSC and the more severe fault they were defined two kinds of short circuit: high impedance short-circuit (HI) and low impedance short-circuit (LI), respectively. The short-circuits are applied in one of the three phases during the data acquisition tests. Three crescent levels of interturn short-circuits are applied both in HI and LI. Represented as the percentage of the total number of stator windings in that phase, the levels are approximately 1.41%, 4.81% and 9.26%. In the following text the HI fault might be referred as HI1, HI2 and HI3 that means HI with 1.41%, with 4.81% and with 9.26% of short circuited windings respectively. The same is valid to LI fault.

### A. Datasets

The two main datasets created are the healthy and the fault datasets, but it is important to emphasize that the fault condition can be subdivided in HI fault and LI fault. The HI and LI can once more be subdivided according the level of interturn short circuit (HI1, HI2, HI3, LI1, LI2, LI3).

The normal condition and each one of the subdivided fault (HI1, HI2, HI3, LI1, LI2, LI3) contains 100,000 time domain samples to each phase. The spectrum in frequency domain is obtained by the Fast Fourrier Transform (FFT).

Due to there is redundant information given by the two phases directly connected with the short circuit, one of them is unnecessary and can be excluded. Thus the total amount of samples is equal to 378 (2 classes x 2 currents x 7 frequencies x 3 loads x 2 kinds of fault x 3 levels of fault). The healthy condition dataset has 42 samples (2 currents x 7 frequencies x 3 loads), and the faulty set has 336 samples, which 168 are HI and 168 LI.

### B. Attributes Selection

The equation (1) is used to make a pre-selection of the harmonics used as ANN attributes. Considering the *s=0* and knowing that *p=2*, the spectrums when *k=1* and *n=1, 2, 3, 4, 5* obtained from (1) are: $0.5f_l$, $1.0f_l$, $1.5f_l$, $2.0f_l$, $2.5f_l$, $3.0f_l$, …. But the slip will never be zero; it depends on the load and also of the drive frequency. So an algorithm is used to get the amplitude of approximated harmonics: those harmonics are used as a central point and a search for the maximum amplitude value around each one of them is done. The range to the search is $\pm s$. Due to the band limit of 500 Hz, it is done until 8th harmonic, which gives 16 pre-selected parameters.

To reduce the number of pre-selected parameters, it was done a statistic analysis of the variance of the pre-selected

[1] Sample Nyquist's Theorem: considering $x_c(t)$ a limited signal in band with its frequency domain version given by $X_c(j\Omega) = 0$ to $|j\Omega| \geq \Omega_N$, then $x_c(t)$ is only determined by its samples $x(n) = x_c(nT), n = 0, \pm 1, \pm 2, \dots$ if $\Omega_s = 2\pi/T \geq 2\Omega_N$. The $\Omega_N$is usually defined as the Nyquist frequency ['OPPENHEIM et al., 1999']

attributes considering all the conditions used (load, frequencies, kind of faults, level of faults). Those with significant values were $0.5f_1$; $1f_1$; $1.5f_1$; $2f_1$; $3f_1$; $5f_1$; $7f_1$. After that it was included $2.5f_1$, $3.5f_1$ because they are the approximated harmonics to a 4 pole motor according [2] that gives information about short-circuit, but after testing the relevancy of each component of these harmonics to the ANNs, the final attributes chosen were $0,5f_1$; $1,5f_1$; $2,5f_1$; $3f_1$; $5f_1$; $7f_1$.

## V. RESULTS

During the training of the ANNs 90% of data samples were used. The 10% remained composed the validation dataset. However, the training dataset was equally divided to each class in a way to avoid tendentious classification. Consequently, due to the healthy dataset having 42 samples and the faulty condition having 336 samples, great part of the faulty dataset are not used during the training, so these data are added to validation dataset.

There are no formulas to select the BP parameters, thus all the parameters used were selected after several attempts. At the end, the hidden-layer was set to 5 neurons. The learning rate used decreases exponentially with the number of epochs until a final value. Also it was used the momentum term. It was implemented the early stop, that uses a test dataset to stop the training when the generalizing error is growing. Thus the datasets were divided in training, testing and validation, respectively set to 70%, 20% and 10% of the total. To the ELM, the only parameter to be adjusted is the number of hidden-neurons that was set to 20 after many experiments.

In both ANNs, the activation functions are hyperbolic tangents and the datasets are normalized by two ways: i) removing the average and dividing by variance and ii) by adjusting the values between -1 and +1.

Evaluate the ANNs by using simple global classification rate does not give information about classification performance of each class, so it is complemented by the use of a confusion matrix CM. The diagonal of the CM has the classification rate to each class. So each line refers to the correct and incorrect classification of one class, totalizing the 100%.

The ANNs and its algorithms are programed using the software MATLAB®. In Table I are shown the average results found after 50 trainings and their standard deviation. Some abbreviations are used: *CR* for classification rate; the subscripts *TR, TS* and *VAL* refer to training, test and validation datasets, respectively; σ for standard deviation; $N_h$ for the number of hidden-neurons; $N_w$ for the total number of weights.

To attest the non-linear separability of the datasets it was included a test with the Simple Perceptron (SP) which is a linear neural classifier. In Table II is shown the Confusion Matrices on Average, $CM$.

The linear classifier Perceptron hits on average about 60% of the training dataset. It means the Perceptron has no capability of mapping the dataset. It also presents poor generalization capability. Looking to MLP results with BP and ELM it is noticeable that the classification to this problem is a hard task even using non-linear classifiers: on average about 65% of the new data are correctly classified both to MLP/BP as to MLP/ELM. The number of weights in MLP/ELM is almost 4 times greater than the MLP/BP, which is an important feature when one thinks in to embed the ANN in micro-processors with limited memory capacity.

Table I. Average results of the ANNs.

| ANN | $N_W$ | $CR_{TR}$ | $\sigma_{TR}$ | $CR_{TS}$ | $\sigma_{TS}$ | $CR_{VAL}$ | $\sigma_{VAL}$ |
|---|---|---|---|---|---|---|---|
| SP | 7 | 60.1 | 20.6 | - | - | 50.5 | 19.2 |
| MLP/BP | 41 | 78.0 | 8.8 | 74.9 | 11.0 | 64.9 | 11.3 |
| MLP/ELM | 161 | 82.5 | 3.7 | - | - | 65.2 | 4.8 |

The MLP/BP presents more equally classification, it is possible to see in $CM_{TR}$ that about 75% of faulty data were correctly classified on average, and about 80% of the healthy were correctly classified. That is a discrepancy of about 5% whereas in the MLP/ELM this discrepancy is about 10%.

Table II. Confusion Matrices on Average.

| | ***MLP/BP*** | | | ***MLP/ELM*** | | |
|---|---|---|---|---|---|---|
| $CM_{TR}$ | | Healthy | Faulty | | Healthy | Faulty |
| | Healthy | 80.21 | 19.79 | Healthy | 87.84 | 12.16 |
| | Faulty | 24.31 | 75.69 | Faulty | 23.00 | 77.00 |
| $CM_{TS}$ | | Healthy | Faulty | - | | |
| | Healthy | 77.45 | 22.55 | | | |
| | Faulty | 27.64 | 72.36 | | | |
| $CM_{VAL}$ | | Healthy | Faulty | | Healthy | Faulty |
| | Healthy | 69.33 | 30.67 | Healthy | 75.60 | 24.40 |
| | Faulty | 35.17 | 64.83 | Faulty | 35.01 | 64.99 |

The faulty dataset is composed by 6 sub-divisions of faults, as early mentioned, named here as HI1, HI2, HI3, LI1, LI2, LI3. Table III shows the average of the correctly classification to each level of fault for the training and the test datasets. It is possible to see that the more the coils are short-circuited, the faults are more correctly classified.

Table III. Percentages of correctly classification to each level of short-circuit.

| **ANN - Dataset** | **HI1** | **HI2** | **HI3** | **LI1** | **LI2** | **LI3** |
|---|---|---|---|---|---|---|
| ***MLP/BP – TR*** | 67% | 72% | 74% | 73% | 84% | 91% |
| ***MLP/ELM – TR*** | 70% | 75% | 77% | 75% | 80% | 92% |
| ***MLP/BP – VAL*** | 59% | 60% | 66% | 64% | 70% | 70% |
| ***MLP/ELM – VAL*** | 60% | 61% | 66% | 64% | 67% | 72% |

Average results are used to evaluate the general behavior of the designed neural networks, but in the real implementation one ANN must be chosen. Therefore one ANN trained by backpropagation and another trained by ELM were chosen to be pruned and then used to show the final results. The pruning method used is called CAPE and it can be found in [19].

The choice of the specifics ANNs take in consideration the classification rate of validation and training datasets, but it was observed mainly the CR to each class. The priority was to choose an ANN that correctly classified all the healthy conditions in the validation datasets. That choice aims to avoid false positives in an online constant monitoring.

In Table IV it is shown that the specific MLP/BP reaches better classification both to training dataset as to validation dataset. Moreover, after pruning the MLP/BP it was able to

improve its generalization capacity. The MLP/ELM was not able to be properly pruned by the CAPE method.

Table IV. Results to specifics ANNs.

| MLP | BP | BP/CAPE | ELM |
|---|---|---|---|
| $N_w$ | 41 | 34 | 161 |
| $CR_{TR}$ | 89.7 | 87.1 | 84.1 |
| $CR_{TS}$ | 81.8 | 81.8 | - |
| $CR_{VAL}$ | 68.5 | 70.2 | 63.8 |
| $CM_{TR}$ | H F<br>H 94.9 5.1<br>F 15.3 84.7 | H F<br>94.9 5.1<br>20.5 79.5 | H F<br>90.2 9.8<br>22.0 78.0 |
| $CM_{TS}$ | H F<br>H 81.8 18.2<br>F 18.2 81.8 | H F<br>81.8 18.2<br>18.2 81.8 | - |
| $CM_{VAL}$ | H F<br>H 100.0 0.0<br>F 32.1 67.9 | H F<br>100.0 0.0<br>39.7 69.3 | H F<br>100.0 0.0<br>36.1 63.9 |

In [1] the same dataset used in this work had been used in a SOM ANN, but with differences in the attributes, and in the normalization. The final result presented by [1] gives a global $CR$ of 87.5%. However, the $CR$ to healthy dataset is 52%, whereas the $CR$ to faulty dataset is 94.5%. That result means the most data are classified as fault, which is a much larger dataset than the healthy one. In an online constant monitoring it also means a higher probability of false positives occurrence.

## VI. CONCLUSIONS

The problem of early fault detection in converter-fed induction motor is a subject that is far to be completely solved. The investigation with real datasets reveals difficulties in separating the faulty datasets from the healthy ones, which reinforces the importance of constant on-line monitoring. The use of converter adds the possibility of to embed the system directly in the equipment, which means more protection to the converter besides all the discussed benefits of the early fault detection for the machine and for industries. The great advantages of this non-invasive detection system make its improvement a task with great potential; one possibility involves the choice of relevant spectrums as parameters to training the classificator which is not an ended issue and is directly related with the classifier accuracy.

The two algorithms used to training the classifier showed similar results, but the ELM computes much faster and is much easier designed, although the MLP/ELM needed four times more neurons in the hidden-layer to do so. The pruning method was capable to improve the generalization in MLP/BP through the removing of connections, but the MLP/ELM was not able to be pruned by the method used.

## ACKNOWLEDGMENT

The authors thank the FUNCAP support and also Oliveira, A. thanks the CAPES for the master's scholarship.

## REFERENCES

[1] Coelho, D.; Medeiros, C., "Short Circuit Incipient Fault Detection and Supervision in a Three-Phase Induction Motor with a SOM-Based Algorithm." Book of Advances in Self-Organizing Maps. vol. 198, p. 315-323. Jan. 2013. ISBN 978-3-642-35229-4.

[2] Thomson, W.T.; Fenger, M.; , "Current signature analysis to detect induction motor faults," Industry Applications Magazine, IEEE , vol.7, no.4, pp.26-34, Jul/Aug 2001.

[3] Nandi, S.; Toliyat, H.A.; Xiaodong Li; , "Condition monitoring and fault diagnosis of electrical motors-a review," Energy Conversion, IEEE Transactions on , vol.20, no.4, pp. 719- 729, Dec. 2005.

[4] Bezesky, D.M.; Kreitzer, S.; , "Selecting ASD systems," Industry Applications Magazine, IEEE , vol.9, no.4, pp. 39- 49, July-Aug. 2003

[5] Kowalski, C.T.; Wolkiewicz, M., "Stator faults diagnosis of the converter-fed induction motor using symmetrical components and neural networks," Power Electronics and Applications, 2009. EPE '09. 13th European Conference on , vol., no., pp.1,6, 8-10 Sept. 2009

[6] Kaufhold, M.; Schäfer, K.; Bauer, K.; Bethge, A. and Risse, J.: "Interface phenomena in stator winding insulation – Challenges in design, diagnosis, and service experience", IEEE Electrical Insulation Magazine, vol. 18, nº 2, pp. 27-36, March/April 2002.

[7] Natarajan, R., "Failure identification of induction motors by sensing unbalanced stator currents," Energy Conversion, IEEE Transactions on , vol.4, no.4, pp.585,590, Dec 1989

[8] Joksimovic, G.M.; Penman, J., "The detection of interturn short circuits in the stator windings of operating motors," Industrial Electronics, IEEE Transactions on , vol.47, no.5, pp.1078,1084, Oct 2000

[9] Stavrou, A.; Sedding, H.; Penman, J., "Current monitoring for detecting interturn short circuits in induction motors," Electric Machines and Drives, 1999. International Conference IEMD '99 , vol., no., pp.345,347, May 1999

[10] Penman, J.; Sedding, H.G.; Lloyd, B.A.; Fink, W. T., "Detection and location of interturn short circuits in the stator windings of operating motors," Energy Conversion, IEEE Transactions on , vol.9, no.4, pp.652,658, Dec 1994.

[11] da S Gazzana, D.; Pereira, L.A.; Fernandes, D., "A system for incipient fault detection and fault diagnosis based on MCSA," Transmission and Distribution Conference and Exposition, 2010 IEEE PES , vol., no., pp.1,6, 19-22 April 2010

[12] Bouzid, M.; Champenois, G.; Bellaaj, N.M.; Signac, L.; Jelassi, K.; , "An Effective Neural Approach for the Automatic Location of Stator Interturn Faults in Induction Motor," Industrial Electronics, IEEE Transactions on , vol.55, no.12, pp.4277-4289, Dec. 2008

[13] Das, S.; Koley, C.; Purkait, P.; Chakravorti, S., "Wavelet aided SVM classifier for stator interturn fault monitoring in induction motors," Power and Energy Society General Meeting, 2010 IEEE , vol., no., pp.1,6, 25-29 July 2010

[14] Thorsen, O.; Dalva, M.; , "Condition monitoring methods, failure identification and analysis for high voltage motors in petrochemical industry," Electrical Machines and Drives, 1997 Eighth International Conference on (Conf. Publ. No. 444) , vol., no., pp.109-113, 1-3 Sep 1997.

[15] Hornik, K.; Stinchcombe, M.; White, H. "Multilayer feedforward networks are universal approximators". Neural Networks, v. 2, p 359-366, 1989.

[16] Guang-Bin Huang; Qin-Yu Zhu; Chee-Kheong Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on , vol.2, no., pp.985,990 vol.2, 25-29 July 2004.

[17] Engelbrecht, A. P. (2007), "Introduction to Computational Intelligence, in Computational Intelligence: An Introduction", Second Edition, John Wiley & Sons, Ltd, Chichester, UK.

[18] Principe JC, Euliano NR, Lefebvre WC (2000) Neural and adaptive systems. Wiley, London.

[19] Medeiros, Cláudio M.S.; Barreto, Guilherme. "A novel weight pruning method for MLP classifiers based on the MAXCORE principle". Journal of Neural Computing and Applications, vol. 22, no 1. 2013. ISSN 0941-0643.