Article

A Non-Invasive Method Based on AI and Current Measurements for the Detection of Faults in Three-Phase Motors

Federico Gargiulo [1,‡] ⓘ *, Annalisa Liccardo [1,‡] ⓘ and Rosario Schiano Lo Moriello [2,‡] ⓘ

1   Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Universitá di Napoli Federico II, 80125 Naples, Italy; federico.gargiulo@unina.it; annalisa.liccardo@unina.it; Tel.: +39-081-768-3912
2   Dipartimento di Ingegneria Industriale, Universitá di Napoli Federico II, 80125 Naples, Italy; rosario.schianolomoriello@unina.it
*   Correspondence: federico.gargiulo@unina.it;
‡   These authors contributed equally to this work.

Abstract: Three-phase motors are commonly adopted in several industrial contexts and their failures [1]
can result in costly downtime causing undesired service outages; this way, motor diagnostics is [2]
an issue that assumes great importance. To prevent their failures and timely face the considered [3]
service outages, a non-invasive method to identify electrical and mechanical faults in three-phase [4]
asynchronous electric motors is proposed in the paper. In particular, a measurement strategy along [5]
with a machine learning algorithm based on Artificial Neural Network is exploited to properly [6]
classify failures. In particular, digitized current samples of each motor phase are first processed by [7]
means of FFT and PSD in order to estimate the associated spectrum. Suitable features (in terms of [8]
frequency and amplitude of the spectral components) are then singled out to either train or feed [9]
a neural network acting as a classifier. The method is preliminary validated on a set of 28 electric [10]
motors, and its performance is compared with common state-of-art machine learning techniques. [11]
The obtained results show that the proposed methodology is able to reach accuracy levels greater [12]
than 98% in identifying anomalous conditions of three-phase asynchronous motors. [13]

Keywords: Failure Prediction; Asynchronous motor; Neural Network [14]

1. Introduction [15]

The three-phase asynchronous motor is widely used as an electric drive thanks to [16]
its design simplicity, low production cost, sturdiness and reliability. Furthermore, the [17]
asynchronous motor is characterized by a high efficiency. It can also be simply connected [18]
directly to the distribution network with constant voltage and frequency if it is not necessary [19]
to control its speed with an inverter. Induction motors are employed in various industries [20]
context and often operate under harsh conditions. Thus, induction motors' internal parts [21]
(such as stator, rotor insulation materials and bearings) can develop faults. Three-phase [22]
electric motors are devices that can therefore show failures due to mechanical and electrical [23]
faults. [1,2]. Induction motors play an important role in industrial manufacturing. To get a [24]
sense about the impact of the induction motors in industrial field, it is worth noting that [25]
these devices account for 29 % of global and 69 % of industrial electricity consumption [26]
[3]. The growing need for online monitoring of the health of systems is necessary to [27]
achieve the high standards of safety and reliability that many industrial contexts require [28]
[4]. Efficient diagnostic procedures help to maintain high levels of Reliability, Availability, [29]
Maintainability, and Safety (RAMS) of the monitored systems [5]. The operating conditions [30]
of the motors typically prevent adequate modeling from an analytical approach, as it would [31]
be necessary to take into account model issues such as workloads, ambient noise and [32]
transient depending on the specific environments and system these motors are exploited [33]
in [6]. Therefore, multiple solutions based on machine learning are emerging in order to [34]
meet the diagnostic needs of the mechanical components that make up devices commonly [35]

adopted in industrial contexts [7,8]. Recent advancements in signal processing and artificial intelligence have attracted renewed interest in induction motor diagnostics thus model-based approaches can be overcome thanks to the machine learning-based fault diagnosis methods [9]. Induction motor faults are mainly diagnosed by using characteristic signals of the motors, such as vibration signals, thermal images, acoustic signals, and motor currents [10]. Unfortunately, most techniques require systems modifications (i.e. sensors installation) or complete observation (as for fault monitoring through infrared sensors). On the contrary, it should be advisable investigating diagnostic solution useful for those production contexts in which the motors could be inaccessible for the application of diagnostic sensors and the only measurable physical quantities are those associated with motor power supply. A promising technique for diagnostics is based on the stator currents analysis which has demonstrated to be useful for preventing catastrophic motor's failures caused by inter-turn short circuit faults in permanent magnet wind generators [11].

In this paper, the issue of motor diagnostics is addressed by means of machine learning and preprocessing techniques applied to current measurements collected from the motor power supply. The proposed method is based on the acquisition of samples of the motor current signals and the use of machine learning techniques based on artificial neural networks for the classification of the motor health status. More specifically, the proposed solution is designed to be implemented in edge computing solutions; to this aim, both techniques of feature extractions and machine learning algorithms have to be as lightweight as possible to be integrated into embedded solutions. Moreover, the non-invasiveness and the high efficacy constitute the novelty of this work; finally, the method performance has been assessed, according to the authors' best knowledge, on the largest dataset of three-phase motors ever presented at the state of the art.

The paper is organized as follows: in Section 2 a State-of-Art overview is reported and recent improvements' limit are discussed; an overview of Artificial Neural Network fundamentals is exploited in 3; the Section 4 presents the proposed method for a non-invasive fault detection based on Current Measurements and AI algorithms; the Section 5 introduces a case study of 28 electrical motors, in Section 5.3 a comparison between the proposed AI algorithms and the most common AI algorithms in predictive maintenance applied is reported. Conclusions and final remarks are highlighted in the Section 6.

2. Related works in Diagnostics for Induction Motors

The progress achieved in the field of motor diagnostics and machine learning techniques is evident at the current state of the art. In recent years, diagnostics of electric motors and health monitoring techniques have become a task of great relevance, as timely maintenance through early detection of irregularities during the normal machine operation can be achieved [12]. Due to the importance of this field, many methods have emerged to carry out diagnostics and predictive maintenance.

New machine learning techniques for prediction tasks are proposed at the state of the art, such as the Non-Iterative Supervised Learning Predictors based on Ito Decomposition and Structure Successive Geometric Transformations Model or the convolutional and Long Short-Term Memory neural networks, generally used for other purposes such as image processing and classification [13–17].

Lingxin Li and Chris Mechefske in the article [18] report a statistics about failure causes. About 50 % of failures are caused by bearing failures, around 40 % are due to winding failures of the stator, the remaining 10 % is due to failures of the rotor or the shafts.

An interesting method for online diagnostics has been presented in [19] where authors proposed the measurement of the frequency response in motor windings in order to perform detection of both mechanical and electrical damage (i.e. deformation of the windings and degradation of conductors and materials for isolation). This method is very interesting as it can be easily integrated into systems based on asynchronous motors and allows for real-time diagnostics. The fact that it doesn't interfere with normal motor operations is its main strength. Unfortunately, the method require a thorough knowledge of the type of

motor being observed, consequently the methodology is difficult to implement on a large scale. Furthermore, the authors do not introduce a quantitative measure of the real ability of the method to identify failures. Finally, the case studies are limited to simulated and not actual faults so there are not enough data to suggest a real applicability.

The methodology proposed in [20] addresses the problem of rotor manufacturing inaccuracies caused during the die casting process. Depending on the importance, this type of problem can present itself immediately or remain hidden until it manifests itself in critical moments of the use of motors. However, this method can only be used offline and is invasive as it is necessary to disassemble the motor.

A smart-sensor has been proposed in [12]; it is based on low cost compact triaxial stray flux sensors whose setup is easy and is non-invasive. Despite this advantages, sensors need to be applied to specific locations on the motors, consequently an offline and invasive intervention on the motor need to be performed.

An interesting comparison between vibrations and current monitoring is presented in [21]. The authors pointed out that methods based on current and vibration analysis are the most widely used techniques for motor diagnostics. This is explained by the advantages of reliability, non-invasiveness, and ease of installation of the measuring sensors. The authors applied the Support Vector Machine algorithm to the different cases of failures and measured signals. With Support Vector Machine it is shown that mechanical failures are better identified by means of vibration signals and electrical failures are well identified by current measurements. It is also pointed out that the accuracy of the diagnostics varies according to the motor speed. Although it is possible to achieve satisfactory accuracy in some contexts, a valid method is not proposed to diagnose every type of fault by means of current signals alone. Furthermore, the authors do not propose a method for the choice of measurement and diagnostic instruments. Finally, the applied method does not seem to be applicable in real time on an embedded device but rather requires an offline computer analysis.

A graph-based semi-supervised learning has been proposed in [9] in order to develop a comprehensive fault diagnosis method for an online diagnostics on induction motors. The proposed method is an approach based on semi-supervised learning which requires a smaller amount of labeled data; in particular, the authors adopted the greedy-gradient max cut algorithm (GGMC). The authors noted that a large labeled dataset is required for supervised machine learning methods, which is not always available in real operating conditions. The method responds to the need to have a consistent training dataset. However, the proposed method has been validated on the same motors used for training, although the data on which the approach has been validated are not the same. Thus there is no information on the validity of the approach in a real context. Finally, the authors do not specify the system requirements to be able to build a diagnostic system for a large-scale application.

3. Fundamentals of Feed-Forward Neural Network

The machine learning technologies adopted in this paper have been appropriately selected for their simplicity, thus making it possible their implementation on embedded solutions and execution on microcontrollers. The proposed method exploits *Feed-Forward* Artificial Neural Network (ANN) as machine learning model. Feed-Forward ANN's connections between the units do not form cycles. In the case of feed-forward networks, function in eq.2 is transformed consequently and it is necessary to introduce the concept of *Neuron* (hence the name) to explain it.

Given an input vector I, a *Neuron* consists of an activation function ϕ that takes as input the weighted sum of the elements of the vector I (fig. 1). This simple neuron is called *Perceptron*, introduced by Rosenblatt in 1962. Usually, a threshold is included in the neuron model, adding a fictitious input with value on this input fixed at 1, the weight of the connection is given by $-t$ that can be imagined as an additional neuron with no input values. Formally the function of the neuron becomes:
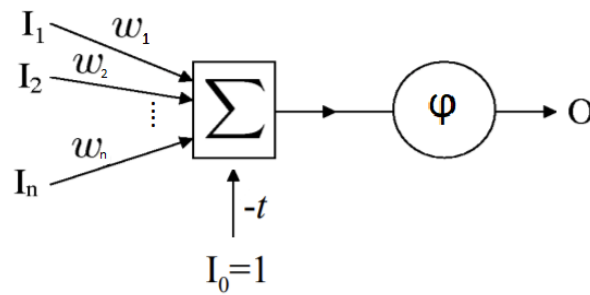
Figure 1. Neuron

$$O = \phi(\sum_{i=0}^{n} \omega_i I_i) \tag{1}$$

where n is the total number of neuron's inputs.

142

(a) ReLU   **(b)** Tanh   **(c)** Sigmoid
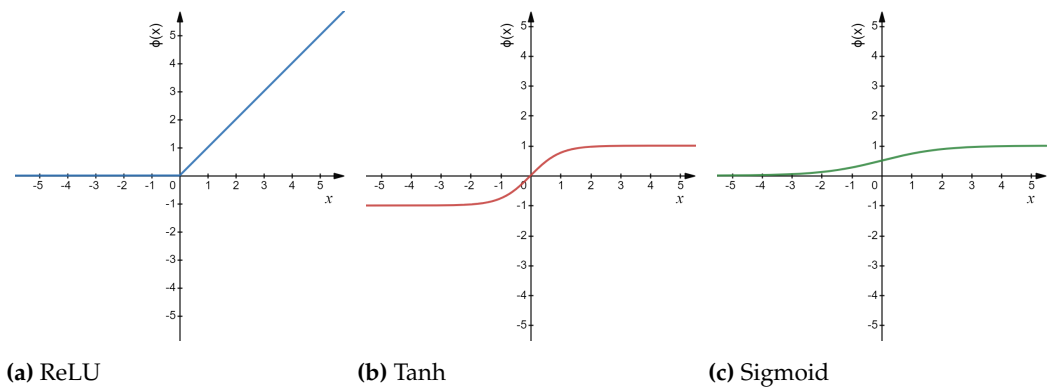
Figure 2. Activation Functions

The activation function ϕ can be of different types, the ones that are adopted in this   143
proposed methodology are listed below.   144

- Rectified Linear Unit (ReLU) which is an activation function defined as the positive   145
  part of its argument (fig. 2a), the function ϕ is $\phi(x) = max(0, x)$;   146
- Hyperbolic tangent (Tanh) which is an activation function defined as $\tanh(x)$ (fig. 2b),   147
  thus the function ϕ becomes $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ 148
- Sigmoid, sometimes named as Logistic or Soft Step (fig. 2c), whose function is $\phi(x) =$ 149
  $\frac{1}{1+e^{-x}}$ 150

It is important to note that in the configuration of the neural network architecture, the   151
activation function is a very important hyperparameter in order to achieve high perfor-   152
mance. All neurons therefore contribute to form the network. If the output of a neuron is   153
linked in input to a new neuron, the latter contributes to form a level that is defined as a   154
deep level. Hence, the network is organized in levels: each neuron of a level receives input   155
only from the neurons of the previous level; it propagates the outputs only towards the   156
neurons of the following levels. The layers between the inputs and the output are called   157
Hidden Layers. Self-connections are not allowed in this type of network, nor are connections   158
between neurons belonging to the same level. Each neuron, therefore, has the function of   159
propagating the signal through the network, with a flow of information that goes from the   160
previous level to the next level (the levels could coincide with the input or output of the   161
network). It follows that the first level of the Neural Networks takes argument of eq. 3 as   162
input.   163
In the fig.3 a generic architecture of a Feed-forward neural network is reported, where   164
$l \in \{2, ..., L\}$ is the layer index, the total number of hidden levels is L and the A_l is the   165

generic number of neurons per level l. The number of intermediate levels $(L-1)$, the num- $_{166}$
ber of neurons for each level (A_l) and the activation functions $(\phi(x))$ are hyperparameters $_{167}$
that compose the architecture configuration and should be established at the first stage, in $_{168}$
order to get the topology, the number and type of neurons, the connections, etc. Further $_{169}$
hyperparameters are added that do not describe the architecture in the strict sense but $_{170}$
contribute to the performance of the model: Regularization strength (Lambda) and The $_{171}$
Standardize data option. The Regularization strength (Lambda) hyperparameter specifies $_{172}$
the regularization penalty term and the Standardize Data binary hyperparameter specifies $_{173}$
whether to standardize the numeric predictors must be standardized or not, in the case of $_{174}$
predictors with widely different scales. The term of the regularization penalty affects the $_{175}$
weight of the regularization; in particular, the risk of overfitting is prevented or increased $_{176}$
according to the regularization value. $_{177}$

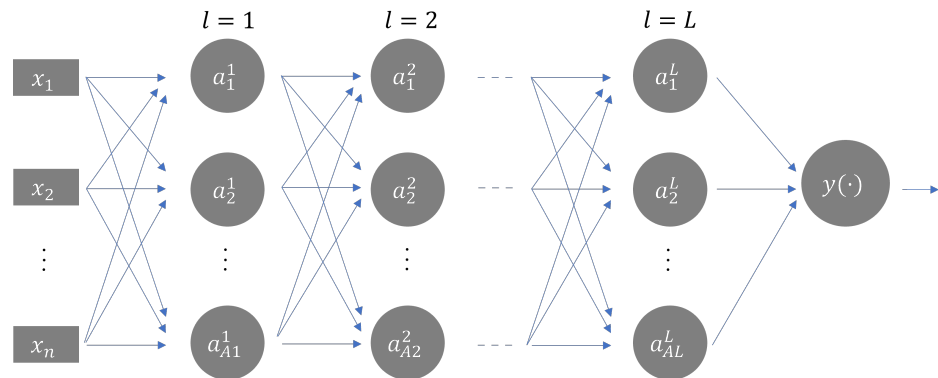**Figure 3.** Feed-forward neural network architecture

Once the architecture is ready, second stage is to determine the weights of the con- $_{178}$
nections in order to build a classifier, based on the training data set made available to the $_{179}$
network (placed in input), this phase is named Training. $_{180}$

4. Proposed Method $_{181}$

Therefore the issue of performing fault detection must be addressed by adopting a $_{182}$
classifier that, on the one hand, ensures a satisfactory level of accuracy in the prediction of $_{183}$
failures and, on the other hand, allows the diagnostic system to be easily integrated into $_{184}$
edge computing solutions. $_{185}$

The goal is therefore to train an algorithm represented by a function $f(X)$ that, given
the current signals measured on the power supply line, returns a symbol that identifies
the class to which the signals belong (healthy / broken). The function to be constructed is
therefore a relationship between a set of cardinality m to a set of cardinality 1. The function
$f(X)$ is therefore the following:

$$Y = f(X) : R^m \rightarrow R^1 \tag{2}$$

where m is the number of features and R^1 is the label set. The function, therefore represent $_{186}$
the classification algorithm and the output is the result of the classification. $_{187}$

The features extracted from the signal are derived from the spectral analysis of the $_{188}$
acquired current signals. The method proposed in this paper does not use time domain $_{189}$
features in order to keep the algorithms independent of the time of observation of the $_{190}$
signals. Naturally, a longer observation time of the signal allows a higher quality of the $_{191}$
samples thanks to the consequent reduction of the noise floor. The method is mainly based $_{192}$
on Fast Fourier Transform (FFT) and Power Spectral Density (PSD) components as they $_{193}$
are easily computable components in embedded solutions such as microcontrollers or $_{194}$
DSPs for which libraries and dedicated hardware are generally available. In particular, $_{195}$
the largest spectral components of the FFT and the largest components of the PSD are $_{196}$
collected from each signal. For each component, amplitude-frequency pairs are selected. $_{197}$

The method proposed hereinafter adopts both FFT and PSD because the aim is to extract a set of characteristics from the signals that are representative of the failure phenomena but yet synthetic to avoid that an excessive amount of information on the signals can cause risks of overfitting. A failure cannot be identified on the basis of thresholds in the FFT or PSD components alone. This is the reason why the proposed method adopts a more complex inferential modeling with a fixed number of features extracted from both the FFT and the PSD.

According to the considered inputs, the function of the classifier therefore becomes:

$$Y = f(\boldsymbol{F}^{FFT}, \boldsymbol{A}^{FFT}, \boldsymbol{F}^{PSD}, \boldsymbol{A}^{PSD}) \quad , \tag{3}$$

where $\boldsymbol{F}$ and $\boldsymbol{A}$ represent frequencies and amplitudes, respectively. The number of components to be selected must be large enough to represent the reconstructed signal with suitable approximation. At the same time it is not advisable to select too large a number of components for reasons of computational complexity and overfitting problems. It should be noted that an excessive number of components (FFT and PSD) risks being representative not only of the useful signal but also of the measurements noise; moreover a large number of features could reduce the model's ability to recognize the same phenomena on typical signals of different motors.
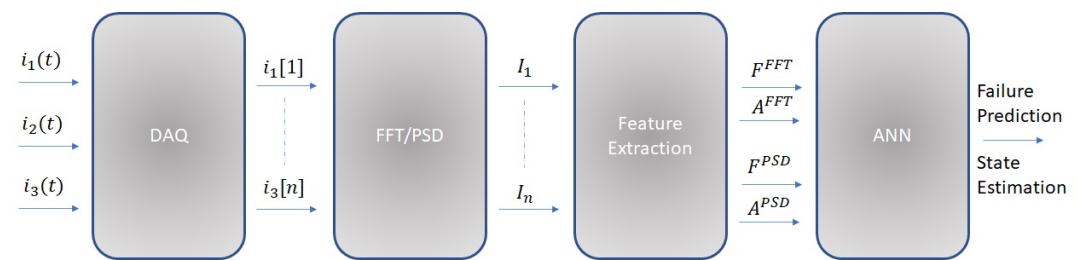

Figure 4. Block Diagram of the proposed Method

The model therefore takes as input the features described above extracted from the observations collected from the supply line.

The Fig. 4 summarizes the method proposed in this paper. A data acquisition unit (DAQ) collects samples from the three-phase motor power supply currents. Metrological characteristics of the DAQ have to be accurately selected in term of both memory depth and sampling rate to accomplish the desired task. Acquired samples are then processed in order to achieve the associated spectral components of interest. Finally, the features are extracted in order to turn them in a suitable way for the ANN training. This process differ in production on the last stage. The trained ANN model is used as classifier and the result represent the electrical motor health state.

However, the choice of the hyperparameters that describe the architecture must be guided by technical approach because the number and the value of hyperparameters combinations can be huge. Of course, the goal is to maximize the classifier performances while trying to minimize the complexity of the architecture.

Hyperparameter configuration can be chosen via grid search based approach, which consists of an exhaustive search in a limited range of possible configurations, or by a random search based approach, which consists of a non-exhaustive and random search for configurations in a range of possible combinations. In case of large number of hyperparameters, the random search technique is to be preferred over the grid search from the computational times' viewpoint, because it has been demonstrated that random search is able to preserve good performance[22].

5. Preliminary performance assessment

The case study on which the method was validated includes a set of motors of various nature and different working conditions.

5.1. Measurement setup

To assess the performance of the methods, a proper measurement station based on an embedded platform has been designed and implemented, In particular, the current sensor chosen for the acquisition is the MCR1101-20-5. The main sensor specification are reported in table 1.

Figure 5. MCR1101-20-5 Package

It was decided to adopt this sensor due to its full scale, passband and limited magnetic hysteresis characteristics. The sensor performance has been assessed in laboratory tests using the Fluke 5720A [23] Calibrator and 5725A Amplifier [24] as reference current sources. The evaluation of the magnetic hysteresis was performed by stimulating the sensor with increasing and decreasing current flows and acquiring ten thousand samples for each current step.

Parameter	Typical Value for $VCC = 5V$ and $T_A = 25°C$
Input Range	$\pm 20\,A$
Sensitivity	$100\,mV/A$
Zero Current Offset	$\pm 20\,mA$
Sensitivity Error	$\pm 0.3\,\%$
Linearity Error	$\pm 0.3\,\%FS$
Total Error	$\pm 0.6\,\%RD$
Zero Current Offset Drift	$\pm 60\,mA$
Sensitivity Drift	$\pm 0.3\,\%$
Total Error Drift	$\pm 0.4\,\%FS$

Table 1. MCR1101-20-5 sensor's characteristics

Obtained results are presented in Table 2; for each value of nominal current I_nom, the averages of 10000 samples for increasing (I_{meas}^+) and decreasing (I_{meas}^-) current flows as well as the respective standard deviation (σ^+ and σ^-) has been reported.

$I_{nom}[A]$	$I_{meas}^+[A]$	σ^+	I_{meas}^-	σ^-	Δ
-10.000	-9.872	0.007	-9.879	0.007	0,007
-9.000	-8.878	0.005	-8.889	0.007	-0.011
-8.000	-7.886	0.006	-7.901	0.006	-0.015
-7.000	-6.926	0.007	-6.911	0.006	0.015
-6.000	-5.913	0.008	-5.912	0.007	0.001
-5.000	-4.926	0.007	-4.942	0.007	-0.016
-4.000	-3.939	0.007	-3.940	0.007	-0.001

-3.000	-2.954	0.007	-2.953	0.007	0.001
-2.000	-1.972	0.008	-1.983	0.008	-0.011
-1.000	-0.978	0.007	-0.980	0.008	-0.002
0.000	0.010	0.008	0.009	0.008	-0.001
1.000	1.009	0.008	1.010	0.007	0.001
2.000	1.990	0.008	2.009	0.007	0.019
3.000	2.988	0.007	3.007	0.008	0.025
4.000	3.986	0.006	3.983	0.007	0.003
5.000	4.992	0.006	4.974	0.007	-0.018
6.000	5.982	0.006	5.970	0.007	-0.012
7.000	6.984	0.005	6.993	0.007	0.009
8.000	7.983	0.006	7.973	0.006	-0.010
9.000	8.961	0.006	8.957	0.006	-0.004
10.000	9.990	0.007	9.990	0.007	0.000

Table 2. Result of the test for the measurement hysteresis assessment

To better appreciate the sensor performance, the difference Δ between increasing and   253
decreasing currents has been provided.   254

Results are also summarized in fig 6, where the evaluation of the differences Δ versus   255
the nominal currents is shown. Intervals centered in the difference Δ, whose half-amplitude   256
is equal to three times the associated standard deviation, are also reported. As it can be   257
noticed all the intervals are metrologically compatible with 0, thus assuring a neglectable   258
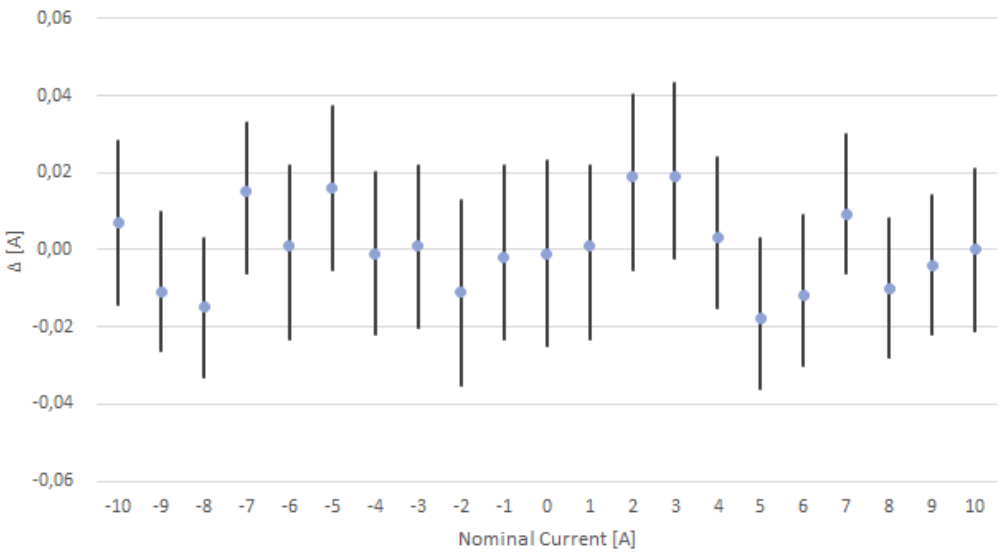contribution of the magnetic hysteresis for the considered application.   259

Figure 6. Magnetic Hysteresis Test

Gain and offset error, equal respectively to -0.838% and $0.290A$, were also evaluated   260
and compensated in the successive processing step.   261
The Microcontroller (MCU) chosen for the measurement setup is the *STM32F4V11VET*. The   262
MCU's characteristics are written below, for the sake of brevity, just information relevant   263
for the case study have been reported below.   264

       265

- Arm® 32-bit Cortex®-M4 CPU with FPU;   266
- 512 Kbytes of Flash memory;   267
- 128 Kbytes of SRAM;   268

- General-purpose DMA;
- Up to 11 timers;
- A 12-bit A/D converter 2.4 MSPS with 16 channels;
- Up to 3 USARTs.

The current sensors output a voltage proportional to the measured current. Since it is necessary to acquire samples coming from 3 motor phases, 3 ADC channels have been used on which the voltage signals coming from the MCR1101-20-5 sensors are input.
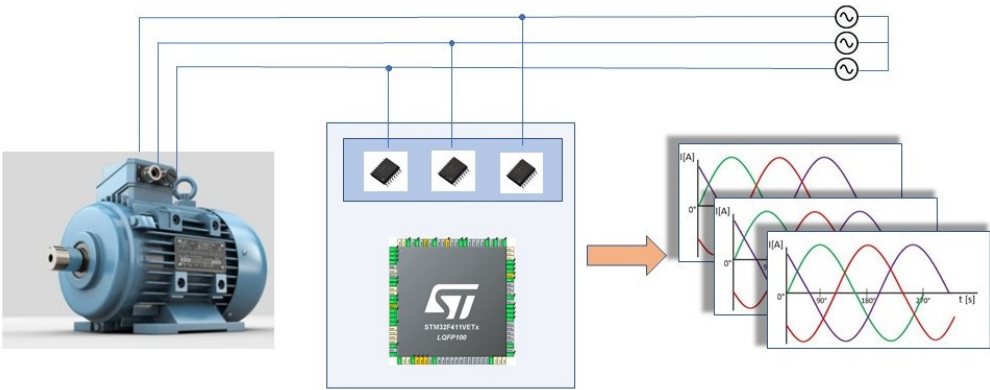

Figure 7. Data Acquisition Schema

It is necessary to reach a sampling rate in order to collect from the three channels measurements at 10000 samples per second. This is possible by using the DMA and setting it so that as soon as the ADC produces a valid value, the DMA takes it to a buffer in RAM. Of course it is necessary to reach a trade off between sample size and available RAM resources.

In this case study it was possible to acquire 20 whole periods with 20000 samples for each phase, for a total buffer of 60000 samples.

The device including sensors and wiring to operate the acquisitions is shown in the figure 8.

5.2. Features extraction and Modelling

To operate the measurement campaign, it was necessary to acquire samples on a large number of motors in different health conditions. Samples from the 3 power supply streams were collected for each motor. The dataset used for the case study is shown in the Table 3.

Classes	Number of Motors	Class'dimension
Healthy	7	21
Faulty	21	63

Table 3. Dataset exploited for the Cross Validation

A large number of spectral components would allow having a complete description of the acquired signal but it would increase the consumption of computing and memory resources for the following steps. A trade-off between the dataset quality and the consumed hardware resources is required.

For each sample, the 10 largest frequency components of the Fast Fourier Transform (FFT) and the 10 largest components of the Power Spectral Density (PSD) were selected.

The dataset is then created by taking the frequencies and amplitudes of the largest 10 components of the FFT and PSD.

$$X = \{f_1, A_1^{FFT}, f_2, A_2^{FFT}, ..., f_{10}, A_{10}^{FFT}, f_1, A_1^{PSD}, f_2, A_2^{PSD}, ..., f_{10}, A_{10}^{PSD}\} \qquad (4)$$
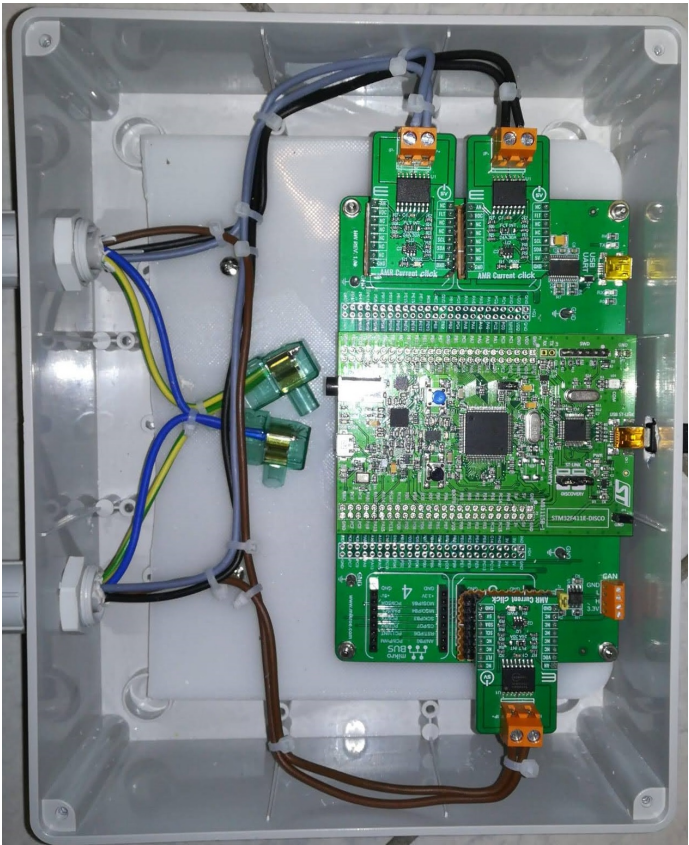
Figure 8. Data Acquisition System

The Dataset is therefore composed of 40 features that describe in a synthetic way, and with a good approximation, the nature of the acquired signal. For the training and testing of the model, the *k-fold* technique was adopted. In this case, 5 folds were selected.

It is necessary to make a choice of hyperparameters before starting the training. As illustrated above, the random search technique can lead to satisfactory results by reducing development times. The table 4 shows the ranges of all the hyperparameters, it is natural that the number of all the possible configurations is very high, this entails a great deal of difficulty in operating a grid search technique (exhaustive evaluation of all configurations).

Hyperparameter	Range
Number of Fully connected Layer	$\{1-3\}$
First Layer Size	$\{1-300\}$
Second Layer Size	$\{1-300\}$
Third Layer Size	$\{1-300\}$
Activation	$\{ReLU; Tanh; Sigmoid\}$
Regularization Strength (Lambda)	$\{1.1905e^{-07}-1190.4762\}$
Standardize Data	$\{Yes-No\}$

Table 4. Hyperparameter configuration - ANN

The training phase is therefore carried out iteratively and it is necessary to introduce criteria with which to terminate it. The iterations can be limited in number, in time or on the basis of an index and the achievement of its threshold value.
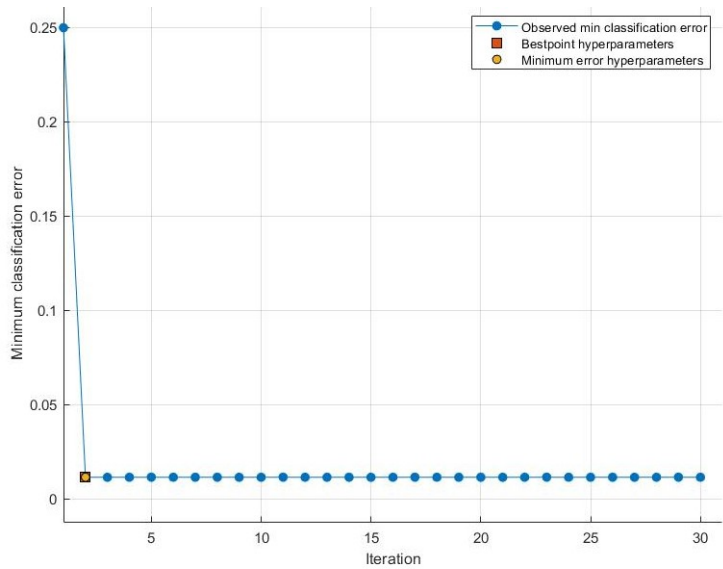
Figure 9. Neural Network Minimum Classification Error

The graph in the plot represents the estimate of the Minimum Classification Error 311
(MCE). The MCE is calculated considering the sets of hyperparameter values for each 312
iteration (blue points). The yellow dot and the red square respectively represent the 313
Minimum Error Hyperparameters and the Bestpoint Hyperparameters. In the figure 9 the 314
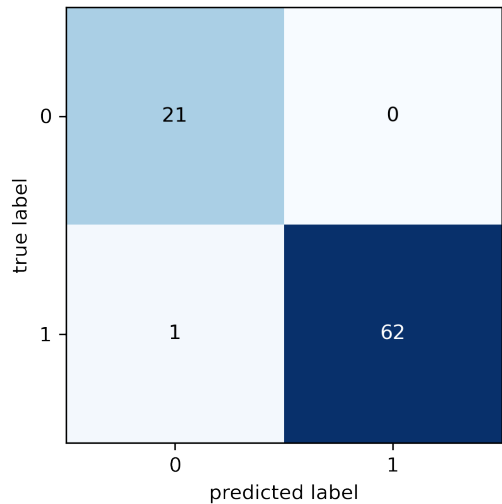Minimum Error Hyperparameters and the Bestpoint Hyperparameters coincide. 315

Figure 10. Neural Network Confusion Matrix

316

The optimized hyperparameters configuration obtained in the case study is reported 317
in the table 5. 318

Hyperparameter	Range
Number of Fully connected Layer	1
First Layer Size	10
Activation	*ReLU*
Regularization Strength (Lambda)	0

Standardize Data	Yes

Table 5. Optimized Hyperparameter Configuration

The results of the cross validation are summarized in the confusion matrix (fig. 10) where known and predicted classes are reported. The false classes (labeled as 0) correspond to observations labeled as healthy, that is, observations collected from the motors in good condition. The true classes are the classes labeled as faulty, i.e. observations corresponding to motors in anomalous conditions (broken bearings, misalignments, etc.). [319][320][321][322][323]

The graph 11 shows the ROC (Receiver Operating Characteristic) Curve which represents the relationship between Sensitivity (True Positive Rate) and Specificity (True Negative Rate). [325][326][327]

The Sensitivity is calculated by taking the ratio between the cases belonging to the class of fault signals correctly classified as positive (the true positives) divided by the sum between true positives and the faulty cases erroneously classified as negative (the false negatives) (5) [25]. This index represents the probability with which a classifier correctly identifies a faulty case as positive.

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad , \quad (5)$$

The Specificity is calculated by taking the ratio between the cases belonging to the class of nominal device signals correctly classified as negative (the true negatives) divided by the sum between true negatives and the healthy cases erroneously classified as positive (the false positives). This index represents the likelihood with which a classifier correctly identifies a healthy case as negative.

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives} \quad , \quad (6)$$

Both sensibility and Specificity indices are calculated from the results presented in the confusion matrix 10. [328][329]
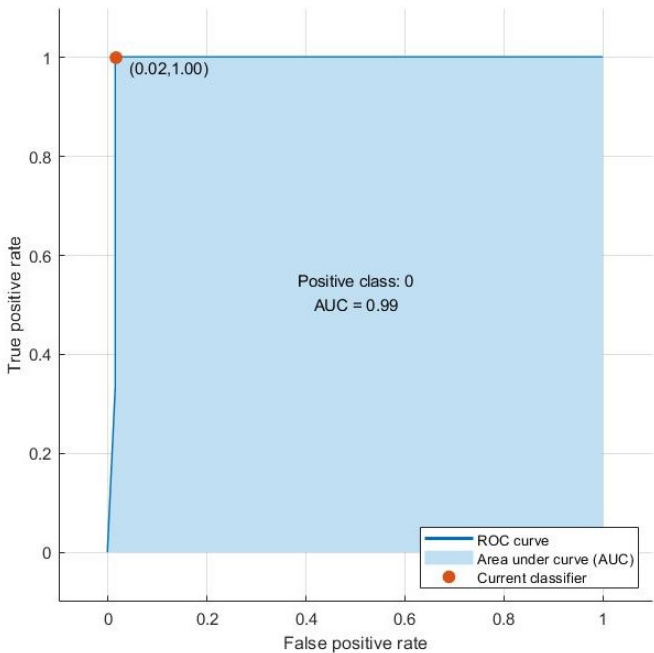

Figure 11. Neural Network ROC Curve

Hyperparameter	Range
Box Constraint level	$\{0.001 - 1000\}$
Kernel scale	$\{0.001 - 1000\}$
Kernel Function	$\{Gaussian, Linear, Quadratic, Cubic\}$
Standardize Data	$\{True, False\}$

Table 6. Hyperparameter configuration - SVM

5.3. Performance Comparison

A comparative analysis was carried out between the results provided by the proposed method and those obtained by replacing the machine learning core with two common classifiers. The selected algorithms for this comparison were chosen for their characteristic of being widely used in diagnostic applications based on the machine learning approach. In particular, the solution based on a feed forward neural network has been compared with Support Vector Machine (SVM) and Decision Tree (DT).

Support Vector Machine is a binary classifier trained on a set of labeled patterns [26]. A Training set can be defined as:

$$(x_i, y_i) \in R^l \times \{\pm 1\} \quad i = 1, ..., N \tag{7}$$

where $x_i \in R^l$ is the input data set and $y_i \in \{\pm 1\}$ is the target. The goal of the support vector machine is to divide the samples by a hyperplane so that the division coincides with the targets y_i.
The classification function is defined as:

$$f(x) = sgn(w \cdot x + b) \tag{8}$$

The function sgn is the bipolar sign function, the vector w is the vector of coefficient and b stands for the bias of the hyperplane.

The classifier hyperplane must be identified in order to satisfy the condition that y_i is greater than or equal to one:

$$y_i[w \cdot x + b] \geq 1, \quad i = 1, ..., N \tag{9}$$

Eq. 9 can be modified as reported in eq. 10, in order to introduce a slack variable to identify a hyperplane that does not fully satisfy the eq. 9 but maximizes the result.

$$y_i[w \cdot x + b] \geq 1 - e_i, \quad i = 1, ..., N \tag{10}$$

The goal of the algorithm is to minimize the following:

$$minJ(w, e, b) = \frac{1}{2}w \cdot w + \frac{1}{2}C \sum_{i=1}^{N} e_i^2 \tag{11}$$

As performed in the case of the neural networks described above, also in this case, for comparative purposes, we proceeded with the evaluation of the performances with the cross validation technique. The random search technique was also applied to the support vector machine for the configuration of the hyperparameters. The possible values that the hyperparameters can assume are shown in the table 6.
Following the application of the random search, the optimal configuration obtained consists of the kernel function such as the Gaussian, the Kernel scale 12.6062, the Box constraint level 167.007, and Standardize data true. This optimal configuration allowed to reach an accuracy of 97.6 %, true positive rate of 100 % and true negative rate of 90.4 %.
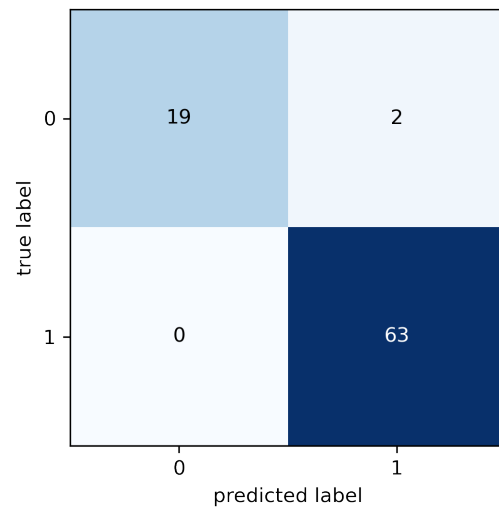
Figure 12. SVM Confusion Matrix

The Support Vector Machine ROC Curve is shown in Figure 14. The curve is quite similar to that obtained with the feed forward neural network proposed in the method, in fact the accuracy level achieved is not much lower. This does not mean that the feed forward neural network is the best choice for predictive maintenance purposes in asynchronous three-phase electric motors.

The Decision Trees are Machine Learning algorithms that can be used for both regression and classification problems. A decision tree is a tree-like model of decisions and it is usually build upside down with its leaves at the bottom.

In decision trees, decisions are represented by the path taken from the root to the leaf node. Tree construction occurs iteratively through leaf splitting or pruning. The random search and cross validation techniques have also been applied in the case of decision trees [27]. The possible values that the hyperparameters can assume are shown in the table 7. It is necessary to establish the optimal split criterion for this case. This choice will also be made by means of the random search technique. Two split criteria are considered: Gini Diversity Index and Maximum Deviance Reduction function. Gini Index (G) is defined according to the formula 12:

$$G = 1 - \sum_k P_k^2 \quad , \tag{12}$$

where the percentage inside a group of elements is defined as P_k and the group of elements must belong to class k [28]. The value G represent the purity and it is equal to 0 if all the elements (inside the group) are part of the same class. Thus, from the branches the node returns as output observation of just one class, given all the elements belong to that specific class, the classification error is null.

The other split criterion is the Maximum Deviance Reduction (MDR) function (sometimes called cross entropy). The function is defined as:

$$MDR = -\sum_k p_k * log_2(p_k) \quad . \tag{13}$$

Also in Maximum Deviance Reduction function, the elements that are part of the class k are represented by the variable p_k which stands for the percentage inside a group [29]. The procedure of splitting keeps going if the conditions are still valid. Of course, a too complex decision tree is not advisable and must be avoided, otherwise there could be risks of overfitting, interpretability and unreliability of predictions.
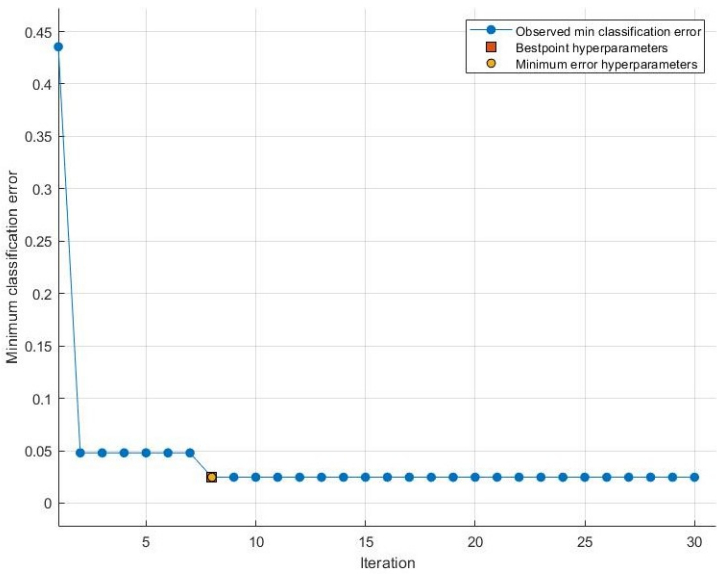
Figure 13. SVM Classification Error

Hyperparameter	Range
Maximum Number of Splits	$\{1 - 83\}$
Split Criterion	$\{Gini's\ diversity\ index;$ $Maximum\ Deviance\ Reduction\}$

Table 7. Hyperparameter configuration - Decision Tree

Following the application of the random search in Decision Tree algorithm, the optimal configuration obtained consists of 6 splits and Gini's diversity index as Split Criterion. This optimal configuration allowed to reach an accuracy of 90.5 %, true positive rate (Sensitivity) of 95.2 % and true negative rate (Specificity) of 76.2 %; the performances are schematically shown in the confusion matrix in fig. 15 and fig. 16.
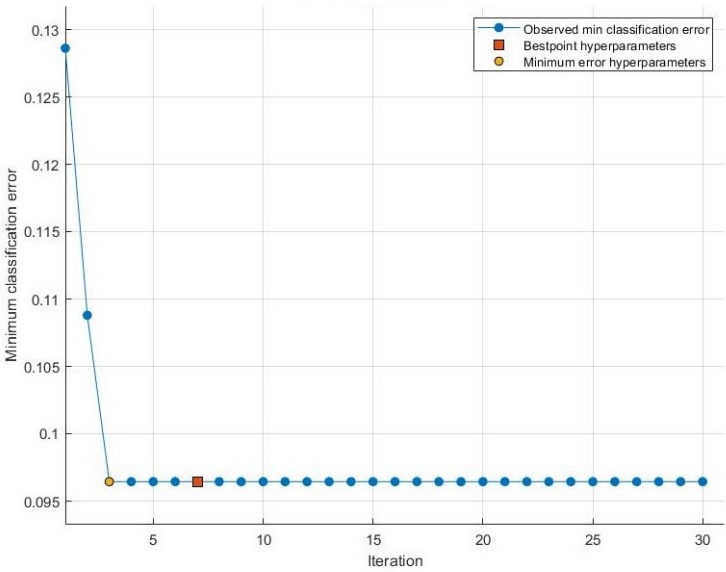
370
371
372
373
374

Figure 16. Decision Tree Classification Error
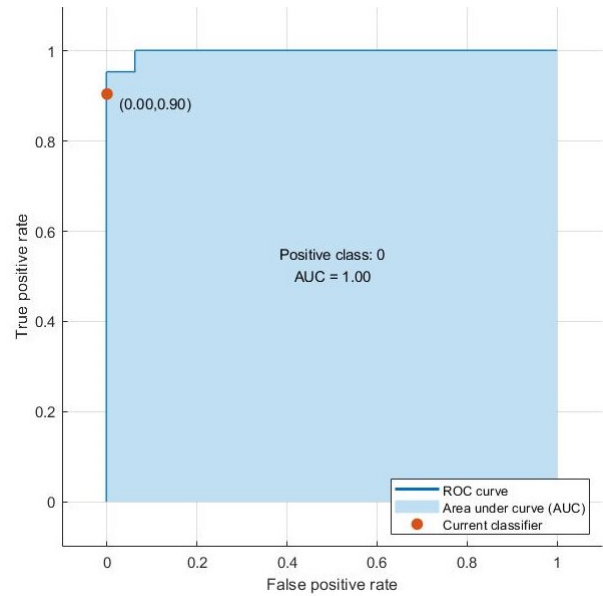
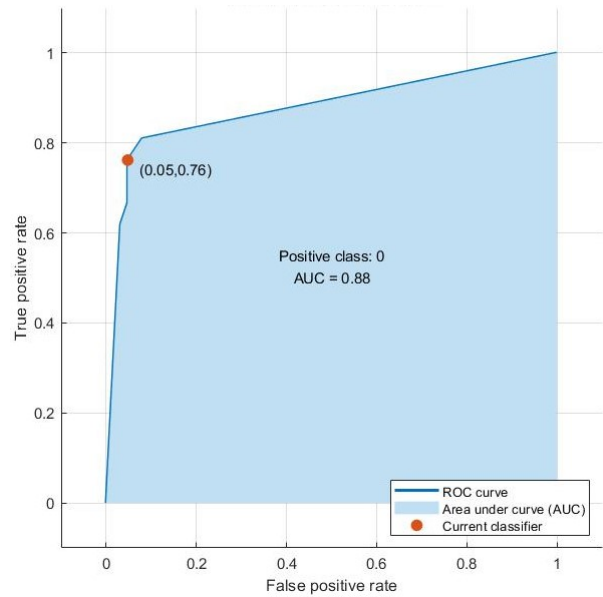**Figure 14.** SVM ROC Curve

Figure 17. Decision Tree ROC Curve

The figure 17 shows the ROC curves of the Decision Tree model. Already graphically it is possible to note that the area under the curve is considerably lower than that of the curves in the two previous models (fig. 11 and fig. 14). This suggests that the performances cannot be superior or equal to those of the other two algorithms previously explored.

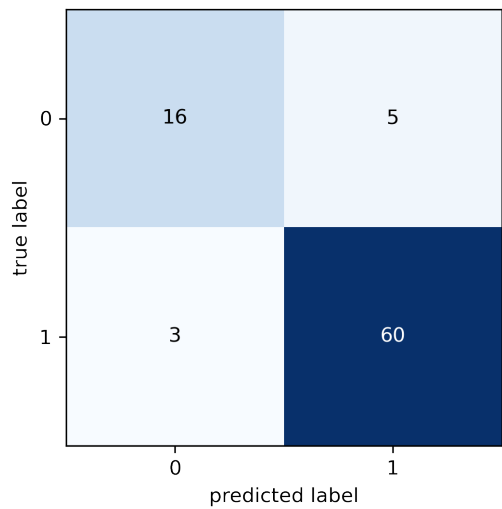| Model	Accuracy	Sensitivity	Specificity
Neural Network Feed-Forward	98.8 %	98.4 %	100 %
SVM	97.6 %	100 %	90.5 %
Decision Tree	90.4 %	95.2 %	76.2 %

Table 8. Performances comparison

Figure 15. Decision Tree Confusion Matrix

5.4. Further Comparisons with State-Of-Art solutions

Additional comparisons, in terms of performance, were carried out by comparing the results published in the literature. An exhaustive comparison is not easy to make as the methods proposed at the state of the art generally have more than one characteristic different from those that make up the method proposed in this work. In order to carry out a suboptimal comparison, all selected works are based on samples coming from the supply current signals.

Authors of [30] have proposed a multi-stage approach based on MLP-ANN machine learning algorithm capable of detecting fault causes in Induction Motors.

Authors of [31] have proposed a method based Frequency plot-based convolutional Neural Network(FOP-CNN) based for detecting motor faults. The study have been performed under different workloads.

Authors of [32] have proposed a method based an unsupervised technique whose advantage is learning from the dataset without an external intervention for data labeling. The machine learning algorithm is based on CNN. The work is focused on bearing faults and no information have been provided regard others kind of faults.

Authors of [33] have been proposed a Empirical Wavelet Transform Convolutional Neural Network (EWT-CNN). The method proposed in the work achieves 97,3% of Accuracy.

All the methods reported in this comparative subsection have been validated on case studies limited to a few units of faulty motors. Moreover, papers considered in tab. 9 do not provide a complete description of the adopted sensors and sample acquisition technologies. This way, it is not possible to hypothesize the absence of overfitting of the trained models.

Method	ML Classifier	Accuracy	Case study motors
Bazan et al. [30]	MLP-ANN	96%	2 Motors
Piedad et al. [31]	FOP-CNN	92.4%	5 Motor
Lei et al. [32]	CNN	99.6%	1 motor
Shao et al. [33]	EWT-CNN	97.3%	6 motors
Proposed method	ANN	98.8%	28 motors

Table 9. Comparison with literature methods based on ML and Current samples.

6. Conclusions

In this paper a method for the predictive maintenance of three-phase asynchronous electric motors has been proposed. The proposed method explores the acquisition techniques for samples of current measures on the supply power lines. The proposed method is based on an analysis carried out on each single phase. For each motor the three phases are used independently in this method. This approach increases the robustness of the method as a problem that can occur on a single phase can be identified by the algorithm. Furthermore, treating the phases separately allows you to triple the size of the dataset, reducing the risk of overfitting.

The preprocessing for feature extraction is also easily implemented in edge computing devices, this allows the implementation and deployment of the proposed method even in real-world contexts where access to external resources is limited. The machine learning algorithm adopted in this method is a classifier based on a Feed-Forward Neural Network. The simplicity of this model is advantageous for edge-computing deployment.

Finally, in this work a further comparison was made between the performance of the feed-forward network-based classifier and other common classifiers in order to demonstrate the highest performance that a feed-forward-based classifier is capable of achieving. It is evident from the experimental data that the proposed method therefore achieves high levels of accuracy (higher than 98%) and a Sensitivity typically greater than 98%.

As for the method limitations, its main weakness is the large dataset required for its training; it would be hard to find such a large number of faulty motors. Furthermore, it is not yet possible to provide information on the nature of the fault affecting the engine to drive maintenance more specifically. This problem is currently under study by focusing on reinforcement learning techniques [34]; results will be presented in the future.

References

1. Martinez-Herrera, A.L.; Ferrucho-Alvarez, E.R.; Ledesma-Carrillo, L.M.; Mata-Chavez, R.I.; Lopez-Ramirez, M.; Cabal-Yepez, E. Multiple Fault Detection in Induction Motors through Homogeneity and Kurtosis Computation. *Energies* **2022**, *15*. doi:10.3390/en15041541.
2. Bandyopadhyay, I.; Purkait, P.; Koley, C. A combined image processing and Nearest Neighbor Algorithm tool for classification of incipient faults in induction motor drives. *Computers & Electrical Engineering* **2016**, *54*, 296–312.
3. Skowron, M.; Orlowska-Kowalska, T.; Wolkiewicz, M.; Kowalski, C.T. Convolutional neural network-based stator current data-driven incipient stator fault diagnosis of inverter-fed induction motor. *Energies* **2020**, *13*, 1475.
4. Chen, Z.; Cao, S.; Mao, Z. Remaining Useful Life Estimation of Aircraft Engines Using a Modified Similarity and Supporting Vector Machine (SVM) Approach. *Energies* **2018**, *11*. doi:10.3390/en11010028.
5. Ciani, L.; Bartolini, A.; Guidi, G.; Patrizi, G. A hybrid tree sensor network for a condition monitoring system to optimise maintenance policy. *Acta IMEKO* **2020**, *9*, 3–9.
6. Shifat, T.A.; Hur, J.W. ANN assisted multi sensor information fusion for BLDC motor fault diagnosis. *IEEE Access* **2021**, *9*, 9429–9441.
7. Arpaia, P.; Cesaro, U.; Chadli, M.; Coppier, H.; De Vito, L.; Esposito, A.; Gargiulo, F.; Pezzetti, M. Fault detection on fluid machinery using Hidden Markov Models. *Measurement* **2020**, *151*, 107126. doi:https://doi.org/10.1016/j.measurement.2019.107126.
8. Teng, W.; Zhang, X.; Liu, Y.; Kusiak, A.; Ma, Z. Prognosis of the Remaining Useful Life of Bearings in a Wind Turbine Gearbox. *Energies* **2017**, *10*. doi:10.3390/en10010032.
9. Zaman, S.M.K.; Liang, X. An effective induction motor fault diagnosis approach using graph-based semi-supervised learning. *IEEE Access* **2021**, *9*, 7471–7482.
10. Lee, J.H.; Pack, J.H.; Lee, I.S. Fault diagnosis of induction motor using convolutional neural network. *Applied Sciences* **2019**, *9*, 2950.
11. Del Pizzo, A.; Di Noia, L.; Lauria, D.; Rizzo, R.; Pisani, C. Stator current signature analysis for fault diagnosis in permanent magnet synchronous wind generators. In Proceedings of the 2015 International Conference on Renewable Energy Research and Applications (ICRERA), 2015, pp. 531–535. doi:10.1109/ICRERA.2015.7418470.
12. Zamudio-RamÃrez, I.; Osornio-RÃos, R.A.; Antonino-Daviu, J.A.; Quijano-Lopez, A. Smart-Sensor for the Automatic Detection of Electromechanical Faults in Induction Motors Based on the Transient Stray Flux Analysis. *Sensors* **2020**, *20*. doi:10.3390/s20051477.

13. Tkachenko, R.; Izonin, I.; Vitynskyi, P.; Lotoshynska, N.; Pavlyuk, O. Development of the Non-Iterative Supervised Learning Predictor Based on the Ito Decomposition and SGTM Neural-Like Structure for Managing Medical Insurance Costs. *Data* **2018**, *3*. doi:10.3390/data3040046.

14. Serradilla, O.; Zugasti, E.; Rodriguez, J.; Zurutuza, U. Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. *Applied Intelligence* **2022**, pp. 1–31.

15. Izonin, I.; Tkachenko, R.; Kryvinska, N.; Tkachenko, P.; et al. Multiple linear regression based on coefficients identification using non-iterative SGTM neural-like structure. In Proceedings of the International Work-Conference on Artificial Neural Networks. Springer, 2019, pp. 467–479.

16. De Santo, A.; Galli, A.; Gravina, M.; Moscato, V.; Sperlì, G. Deep Learning for HDD health assessment: An application based on LSTM. *IEEE Transactions on Computers* **2020**, *71*, 69–80.

17. Wang, L.; Xu, X.; Dong, H.; Gui, R.; Yang, R.; Pu, F. Exploring convolutional LSTM for PolSAR image classification. In Proceedings of the IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2018, pp. 8452–8455.

18. Li, L.; Mechefske, C.K. Induction motor fault detection & diagnosis using artificial neural networks. *International Journal of COMADEM* **2006**, *9*, 15.

19. Bucci, G.; Ciancetta, F.; Fiorucci, E. Apparatus for Online Continuous Diagnosis of Induction Motors Based on the SFRA Technique. *IEEE Transactions on Instrumentation and Measurement* **2019**, *69*, 4134–4144.

20. Kacor, P.; Bernat, P.; Moldrik, P. Utilization of Two Sensors in Offline Diagnosis of Squirrel-Cage Rotors of Asynchronous Motors. *Energies* **2021**, *14*. doi:10.3390/en14206573.

21. Gangsar, P.; Tiwari, R. Comparative investigation of vibration and current monitoring for prediction of mechanical and electrical faults in induction motor based on multiclass-support vector machine algorithms. *Mechanical Systems and Signal Processing* **2017**, *94*, 464–481.

22. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research* **2012**, *13*.

23. Chen, S.F.; Amagai, Y.; Maruyama, M.; Kaneko, N.h. Uncertainty evaluation of sampling measurement system using AC-programmable Josephson voltage standard. In Proceedings of the 29th Conference on Precision Electromagnetic Measurements (CPEM 2014). IEEE, 2014, pp. 258–259.

24. Angrisani, L.; Bonavolontà, F.; Liccardo, A.; Schiano Lo Moriello, R.; Serino, F. Smart power meters in augmented reality environment for electricity consumption awareness. *Energies* **2018**, *11*, 2303.

25. Gargiulo, F.; Duellmann, D.; Arpaia, P.; Schiano Lo Moriello, R. Predicting Hard Disk Failure by Means of Automatized Labeling and Machine Learning Approach. *Applied Sciences* **2021**, *11*. doi:10.3390/app11188293.

26. Sun, B.y.; Lee, M.c. Support vector machine for multiple feature classifcation. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo. IEEE, 2006, pp. 501–504.

27. Zhong, Y. The analysis of cases based on decision tree. In Proceedings of the 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016, pp. 142–147. doi:10.1109/ICSESS.2016.7883035.

28. Tangirala, S. Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm. *International Journal of Advanced Computer Science and Applications* **2020**, *11*, 612–619.

29. Popova, O.; Popov, B.; Karandey, V.; Gerashchenko, A. Entropy and algorithm of obtaining decision trees in a way approximated to the natural intelligence. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* **2019**, *13*, 50–66.

30. Bazan, G.H.; Goedtel, A.; Duque-Perez, O.; Morinigo-Sotelo, D. Multi-Fault Diagnosis in Three-Phase Induction Motors Using Data Optimization and Machine Learning Techniques. *Electronics* **2021**, *10*. doi:10.3390/electronics10121462.

31. Piedad, E.J.; Chen, Y.T.; Chang, H.C.; Kuo, C.C. Frequency Occurrence Plot-Based Convolutional Neural Network for Motor Fault Diagnosis. *Electronics* **2020**, *9*. doi:10.3390/electronics9101711.

32. Lei, Y.; Jia, F.; Lin, J.; Xing, S.; Ding, S.X. An Intelligent Fault Diagnosis Method Using Unsupervised Feature Learning Towards Mechanical Big Data. *IEEE Transactions on Industrial Electronics* **2016**, *63*, 3137–3147. doi:10.1109/TIE.2016.2519325.

33. Shao, H.; Jiang, H.; Zhang, X.; Niu, M. Rolling bearing fault diagnosis using an optimization deep belief network. *Measurement Science and Technology* **2015**, *26*, 115002.

34. Maree, C.; Omlin, C. Reinforcement Learning Your Way: Agent Characterization through Policy Regularization. *AI* **2022**, *3*, 250–259. doi:10.3390/ai3020015.