

Unsupervised Electric Motor Fault Detection by Using Deep Autoencoders

Emanuele Principi, Damiano Rossetti, Stefano Squartini, *Senior Member, IEEE*,
and Francesco Piazza, *Senior Member, IEEE*

Abstract—Fault diagnosis of electric motors is a fundamental task for production line testing, and it is usually performed by experienced human operators. In the recent years, several methods have been proposed in the literature for detecting faults automatically. Deep neural networks have been successfully employed for this task, but, up to the authors' knowledge, they have never been used in an unsupervised scenario. This paper proposes an unsupervised method for diagnosing faults of electric motors by using a novelty detection approach based on deep autoencoders. In the proposed method, vibration signals are acquired by using accelerometers and processed to extract Log-Mel coefficients as features. Autoencoders are trained by using normal data only, i.e., data that do not contain faults. Three different autoencoders architectures have been evaluated: the multi-layer perceptron (MLP) autoencoder, the convolutional neural network autoencoder, and the recurrent autoencoder composed of long short-term memory (LSTM) units. The experiments have been conducted by using a dataset created by the authors, and the proposed approaches have been compared to the one-class support vector machine (OC-SVM) algorithm. The performance has been evaluated in terms area under curve (AUC) of the receiver operating characteristic curve, and the results showed that all the autoencoder-based approaches outperform the OC-SVM algorithm. Moreover, the MLP autoencoder is the most performing architecture, achieving an AUC equal to 99.11%.

Index Terms—Autoencoder, convolutional neural networks, electric motor, fault detection, long short-term memory, neural networks, novelty detection.

I. INTRODUCTION

ELectric motors are used in several everyday devices, such as computers hard drives, washing machines, mixers, fans, and electric cars [1]. Given their wide diffusion, it is essential to diagnose possible faults and non-idealities at the initial stage of their production. Indeed, early detection of faulty units is a topic of major importance, since their failure can cause dangerous situations for people and great economic losses. Electric motors are produced in large quantities and, in order to ensure low production costs, it is difficult to guarantee

Manuscript received October 17, 2018; revised December 1, 2018 and December 19, 2018; accepted January 13, 2019. Recommended by Associate Editor Zhanshan Wang. (*Corresponding author: Emanuele Principi.*)

Citation: E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 441–451, Mar. 2019.

E. Principi, S. Squartini, and F. Piazza are with the Department of Information Engineering, Università Politecnica delle Marche, Ancona 60121, Italy (e-mail: {E.principi; s.squartini; f.piazza}@univpm.it).

D. Rossetti is with Loccioni Group, Angeli di Rosora, Ancona 60121, Italy (e-mail: lordrox@hotmail.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2019.1911393

all the necessary quality tests for each unit produced. Testing is time-consuming and requires proper structures and equipments, making it difficult to integrate it within a production line.

A common solution for the end-of-line quality control is to rely on the experience of qualified operators who manually check the status of the motors. However, this type of test is characterized by little repeatability as the evaluation is influenced by the operator's sensitivity and perception, by their psycho-physical state, and by possible disturbances in the environment (e.g., background noise). In order to ensure a more reliable quality control and to test each element produced by the line, it is necessary to replace the human operator's control with a system able to test automatically the component in a limited time interval to impact as less as possible on the production time of the line.

Due to the importance of the subject, several solutions have been presented in the literature for fault detection and quality control of electric motors [1]–[7]. The proposed methods can be divided in five main categories: model-based, signal-based, knowledge-based, hybrid, and active fault diagnosis methods [3], [4].

Model-based methods use mathematical models that describe motor behaviour by using the physical relationship between the stimuli applied to the motor and its responses [3], [8]. The main problem of this approach is the a-priori knowledge required to describe the motor with the proper mathematical model. Electric motors are complex systems, subject to many variables that change their characteristics over time due to wear and changing operating conditions. It is therefore difficult to find suitable relationships between the available parameters and the motor status that can be used for fault detection.

Signal-based fault diagnosis methods extract significant features from a measured signal and detect faults by using prior knowledge on non-faulty motors [3], [9]. Features can be extracted in the time-domain, in the frequency domain or in the time-frequency-domain. Similarly to model-based methods, these approaches need an a-priori knowledge on faults and health characteristics, that may need a specific manual tuning depending on the operating conditions.

The third class, knowledge (or data-driven) based methods, uses directly data extracted from the motor to characterize its state. These methods use signal analysis and machine-learning techniques to train expert systems able to detect faulty motors. Training of the system uses signals measured from the device under test and it needs the knowledge of the motor condi-

tion, faulty or non-faulty, that is derivable from evaluations performed by human operators. Data-driven methods can be further divided in supervised [5], [10]–[12] and unsupervised [13]–[24]: the first methods require a training dataset where faults are explicitly labelled. This requirement can be difficult to satisfy, since it is necessary to observe a-priori all the possible faults that may occur during the lifetime of an electric motor. Moreover, manual labelling is still a time-consuming and error-prone task. Unsupervised approaches, on the other hand, relax this requirement since their operating principle is the detection of occurrences that deviate from a normal, i.e., non-faulty, behaviour. Thus, only non-faulty data are needed to properly design the fault detection algorithm. These methods will be discussed in more detail in Section II.

The fault detection method proposed in this paper belongs to the family of data-driven unsupervised approaches. More in details, motor faults are detected by acquiring vibration signals and processing them to extract the features necessary to discriminate faults from non-faults. A deep autoencoder network is then used for discriminating between normal and faulty data. Up to the authors' knowledge, this is the first time that an unsupervised method based on deep neural networks is used for detecting motor faults from vibration signals.

The outline of the paper is the following: Section II illustrates the recent works recently presented in the literature that address motor fault detection with knowledge-based methods and presents the contribution of this paper. Section III describes in details the proposed method. Section IV presents the one-class support vector machine (OC-SVM) approach that has been compared to the proposed solution. Section V illustrates the experimental procedure and the obtained results. Finally, Section VI concludes the paper and presents future developments.

II. RELATED WORKS AND CONTRIBUTION

As aforementioned, knowledge-based fault detection approaches can be divided in two classes: supervised and unsupervised. Among supervised approaches, in [11] the authors presented a method for analysing the frame vibrations of a three-phase induction motor during start-up by using the continuous wavelet transform and support vector machine (SVM) classifier. Ghate and Dudul [12] measured the stator current and used a radial-basis-function multi-layer perceptron (MLP) for detecting faults of a three-phase induction motor, in particular stator winding interturn short and rotor eccentricity faults. In [25], the authors have used autoencoders for pre-training a neural network that processes spectral vibration data. Ince *et al.* [26] applied 1-D convolutional neural networks (CNNs) directly to the raw motor current signal, thus eliminating the need for a separate feature extraction stage. The approach was evaluated for detecting bearing faults and demonstrated its superiority compared to conventional feature extraction methods. CNNs have been also applied in [27] for multiclass classification of bearing faults. As in [26], the network processes directly raw vibration signals and extracts features by using wide kernels in the first convolutional layer. Neural networks have been used also in [5], where the authors

proposed a fault classification method based on convolutional neural networks that process directly the raw vibration signal. More in details, the input signal is converted in a 2-D matrix by stacking zero-padded portions of the time sequence and then is processed by using the LeNet-5 CNN [28]. The algorithm proposed in [29] integrates the support vector data description framework in the conventional multiclass SVM for broken rotor bar fault detection in induction motors. The measured signal is the motor current, from which the stationary wavelet packet transform is calculated and used as feature. Sun *et al.* [30] combined compressed sensing and autoencoder networks for bearing fault diagnosis. In particular, an autoencoder is trained for extracting features from vibration signals that then are used by a neural network for supervised fault classification. A similar approach has been presented in [31], where a normalized sparse autoencoder is used for extracting features from raw signals, and a neural network denoted as local connection network is used to classify the fault type. Shao *et al.* [32], used an ensemble of autoencoders with different activation functions for feature learning and fault diagnosis.

Regarding unsupervised approaches, Soualhi *et al.* [14] proposed the artificial ant clustering technique to detect faults by measuring the current and voltage of a squirrel cage induction motor. As features, they used statistical descriptor such as standard deviation and mean values of the current, and active and reactive power signals, spectrum features from the current signal, power factor and impedance. Cho *et al.* [15] trained a set of neural networks to model the faults of an electric motor. One network models the normal behaviour of the motor, while each additional network models a single fault condition. Despite each network represents a model of a normal or faulty system, the approach can be considered knowledge-based since models are directly obtained from data. Moreover, models of faulty conditions are used only for fault isolation, while detection uses only non-faulty data. Razavi-Far *et al.* [13] evaluated several one-class classifiers for detecting broken rotor bar in induction motors. Detection is based on vibration signals, and for classification they studied density-based classifiers, nearest neighbour and k -nearest neighbour, angel-based outlier factor, and k -means. The paper concludes that k -nearest neighbour is the most performing among the evaluated approaches.

As evidenced in the examined literature, neural networks have been widely used in supervised approaches, but scarcely in unsupervised ones. Approaches based on autoencoders such as [30]–[32] train them without supervision for extracting features from the raw vibration signals, but then train the final classifier in a supervised way. Up to the authors' knowledge, the only solution is [15], which however is a hybrid knowledge-based/model-based approach. The main contribution of this paper is the study of neural networks autoencoders as a method for detecting motor faults without supervision. More in details, the approach follows the reconstruction-based novelty detection paradigm [17], where a model is trained to reconstruct fault-free (i.e., normal) data with low error and faulty data with higher error. The magnitude of the error is then used to detect whether the input data belongs to a faulty motor or not. For reconstructing the input data, in this

paper we use deep neural networks autoencoders, i.e., neural networks trained to reconstruct their inputs. Three different types of autoencoders are evaluated: the MLP autoencoder, the CNN autoencoder, and a recurrent autoencoder composed of long short-term memory (LSTM) units. The autoencoders take as input Log-Mel coefficients, which are extracted from the measured vibration signals. The three architectures have been evaluated by using a dataset created by the authors and compared to the OC-SVM [33]. The results show that the proposed neural networks-based algorithm outperforms the OC-SVM approach, and that the MLP autoencoder provides the overall highest performance.

III. PROPOSED METHOD

The block scheme of the proposed method is shown in Fig. 1. Both in the training and in the detection phase, the input signal is processed by the feature extraction stage, where Log-Mel coefficients are calculated. Log-Mels are then processed by a deep autoencoder, a neural network that is trained to reconstruct its input [34]. During the training phase, the autoencoder is trained by using only normal data, i.e., sequences that do not contain faults (Fig. 1(a)). In the detection phase, the autoencoder takes as input the features extracted in the feature extraction stage, but this time the signal can contain a fault or not. In the first case, the reconstruction error is expected to assume considerably higher values than the second case. The “Decision” stage marks the input as a “Fault” if the reconstruction error exceeds a certain threshold, otherwise as “Non-Fault”. The remainder of this section describes in detail the stages of the algorithm.

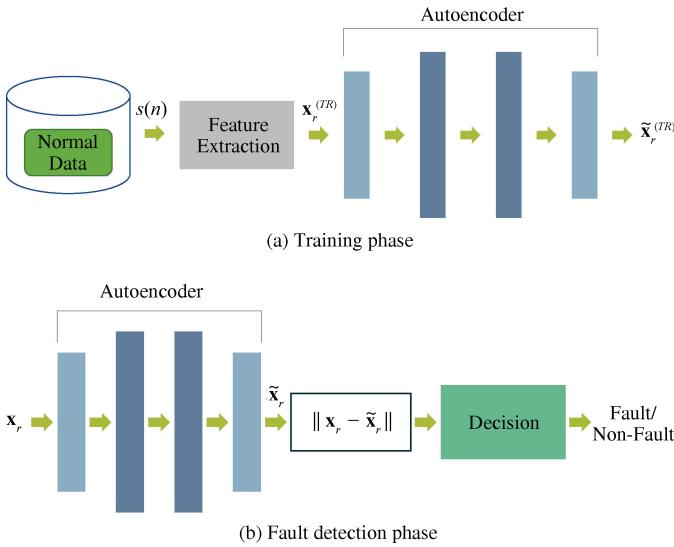


Fig. 1. Block scheme of the proposed approach. In the training phase (a), the superscript (TR) distinguishes a feature vector of the training phase from the one of the test phase shown in (b).

A. Feature Extraction

The feature extraction stage reduces the dimensionality of the input data, while retaining significant characteristics that allow to discriminate whether an input sequence contains a fault or not. In this paper, we used Log-Mel coefficients, a

set of spectral features that have been widely used for speech applications [35], [36], but that have also been successfully applied to the analysis of vibration signals [37], [38].

The first step for extracting Log-Mel coefficients is dividing the input signal in frames of length 600 ms and overlapped by 300 ms (i.e., 50%). Overlapping two consecutive frames by 50% results in a good compromise between time-resolution and computational requirements of the algorithm.

Denoting with $s(n)$ the input signal and with $w(n)$ the window function, the r -th frame $s_r(m)$ of $s(n)$ is defined as:

$$s_r(m) = s(rR + m)w(m), \quad -\infty < r < \infty, \quad 0 \leq m \leq L - 1 \quad (1)$$

where R is the frame step and L is the length of the window $w(n)$. A common choice for the window function is the Hamming window, whose form is the following [39]:

$$w(n) = \begin{cases} 0.54 - 0.46 \cos \frac{2\pi n}{L}, & 0 \leq n \leq L - 1 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The second step from extracting Log-Mels is calculating the N -points discrete Fourier transform of $s_r(m)$:

$$S_r(k) = \sum_{m=0}^{L-1} s_r(m) e^{-j \frac{2\pi}{N} km} \quad (3)$$

where k denotes the frequency bin.

Log-Mel coefficients are obtained by filtering $S_r(k)$ with a filterbank composed of B triangular filters equally spaced along the mel-scale, and then calculating the logarithm of the energy in each band. The b -th triangular filter $h_b(k)$ of the filterbank is calculated as follows [40]:

$$h_b(k) = \begin{cases} 0, & k < f_{b-1}, \\ \frac{k - f_{b-1}}{f_b - f_{b-1}}, & f_{b-1} \leq k < f_b, \\ \frac{f_{b+1} - k}{f_{b+1} - f_b}, & f_b \leq k < f_{b+1}, \\ 0, & k \geq f_{b+1}, \end{cases} \quad b = 1, 2, \dots, B \quad (4)$$

where the boundary frequencies f_b are obtained as:

$$f_b = \left(\frac{N}{f_s} \right) \text{Mel}^{-1} \left(\text{Mel}(f_{low}) + b \cdot \frac{\text{Mel}(f_{high}) - \text{Mel}(f_{low})}{B+1} \right), \quad (5)$$

with

$$f_0 = \left(\frac{N}{f_s} \right) f_{low}, \quad f_{B+1} = \left(\frac{N}{f_s} \right) f_{high}, \quad (6)$$

f_s is the sampling frequency, and f_{low} and f_{high} are respectively the lowest and highest frequencies of the filterbank. The transformation $\text{Mel}(\cdot)$ and its inverse $\text{Mel}^{-1}(\cdot)$ respectively map a linear frequency into the mel scale and vice-versa, and they are defined as follows [41]:

$$\text{Mel}(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right), \quad (7)$$

$$\text{Mel}^{-1}(f) = 700 \left[10^{\frac{f}{2595}} - 1 \right]. \quad (8)$$

A single Log-Mel coefficient is then given by:

$$x_r(b) = \log \left(\sum_{k=0}^{N-1} h_b(k) |S_r(k)|^2 \right). \quad (9)$$

Since the filterbank is composed of B filters, the final feature vector is structured as follows:

$$\mathbf{x}_r = [x_r(1), \dots, x_r(B)]^T. \quad (10)$$

The actual number of filters B will be specified in Section V based on the analysis of the bandwidth of the vibration signals.

B. Neural Network

The topology adopted for detecting faults is the autoencoder, a neural network trained to reconstruct its input [34]. This approach for detecting abnormal data belongs to the family of reconstruction-based methods [16], [17], where an algorithm is expected to reconstruct normal data with low error and abnormal data with a higher error. The magnitude of the error is then used to decide whether the input data is normal or not. In this paper, we evaluated an MLP, a CNN, and a recurrent autoencoder, in particular composed of LSTM units.

1) Multi-layer Perceptron (MLP) Autoencoder: An autoencoder is usually viewed as a network composed of two parts: an encoder followed by a decoder. For the sake of simplicity, consider a neural network composed of a single hidden layer. The encoder maps the input vector $\mathbf{x} \in \mathbb{R}^B$ in the hidden representation $\mathbf{h} \in \mathbb{R}^P$ as follows

$$\mathbf{h} = f(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \quad (11)$$

where $f(\cdot)$ is a component-wise non-linear activation function, $\mathbf{W}_1 \in \mathbb{R}^{P \times B}$ is a weight matrix, $\mathbf{b}_1 \in \mathbb{R}^P$ is a bias vector, and the frame index r has been omitted for simplicity of notation. The decoder network, then, tries to reconstruct the input \mathbf{x} by using the following expression

$$\tilde{\mathbf{x}} = f(\mathbf{W}_2 \mathbf{h} + \mathbf{b}_2), \quad (12)$$

where $\mathbf{W}_2 \in \mathbb{R}^{B \times P}$ is the decoder weight matrix, and $\mathbf{b}_2 \in \mathbb{R}^B$ is the decoder bias vector. The entire autoencoder network, thus, performs the following operation:

$$\tilde{\mathbf{x}} = f(\mathbf{W}_2 f(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \quad (13)$$

Training of the autoencoder is performed by using a dataset composed of normal data only and by minimising the mean squared error (MSE) loss function:

$$L(\mathbf{w}) = \|\tilde{\mathbf{x}} - \mathbf{x}\|^2, \quad (14)$$

where \mathbf{w} contains all the network parameters, i.e., the elements of \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{b}_1 , and \mathbf{b}_2 . The difficulty of training an autoencoder is avoiding the network to learn to simple copy the input to the output. The scientific literature is rich of methods for avoiding this problem [34]. In the proposed approach, the adopted technique consists in corrupting the input data \mathbf{x} with a Gaussian noise with zero mean and σ^2 variance. Since the reconstruction target is the uncorrupted vector \mathbf{x} , the network must learn to reconstruct the input and to remove the additive noise. This procedure proved to be successful for implicitly learning the structure of the data distribution [34].

The autoencoder described previously is composed of two layers, one for the encoder and one for the decoder, however their number can be increased to create a deep autoencoder. The actual topology of the network will be determined in the experiments described in Section V by using a validation set. As activation function $f(\cdot)$, we used the hyperbolic tangent non-linearity. Moreover, batch normalisation [42] has been applied after all the layers, excepting the last.

2) Convolutional Neural Network (CNN) Autoencoder: The CNN autoencoder is composed of an encoder with one or more convolutional layers followed by a pooling operation, and one or more fully connected layers. Symmetrically, the decoder is composed of one or more fully connected layers followed by convolutional layers and upsampling layers. Differently from the fully connected autoencoder described previously, the CNN autoencoder operates on a sequence of R features vectors, instead of a single one. For the sake of simplicity, consider an encoder and a decoder with one convolutional layer and one fully connected layer as in Fig. 2. In the encoder, the convolutional layer operates on an input matrix $\mathbf{X} \in \mathbb{R}^{B \times R}$, where R is the number of frames of a sequence. The output of the layer is an $B \times R \times C$ tensor \mathbf{H} whose elements are calculated as follows:

$$H(i, j, c) = f \left(\sum_{m=0}^{k_i-1} \sum_{r=0}^{k_j-1} X(i+m, j+r) K_c(m, n) \right) \quad (15)$$

where $K_c(\cdot, \cdot)$ is a two-dimensional kernel of size $k_i \times k_j$, C is the number of kernels, $f(\cdot)$ is the non-linear activation function, and $X(i, j)$ is the element at row i and column j of matrix \mathbf{X} . Note that the first two dimensions of \mathbf{H} are the same of \mathbf{X} due to zero-padding. Moreover, the actual operation performed in the above equation is a cross-correlation, but it is usually called “convolution” in the machine learning community [43].

Convolutional layers are often followed by a pooling operation that reduces the dimensionality of the input. In this paper, we used the max-pooling operation which calculates the maximum value over a $p \times p$ window:

$$\tilde{H}(i, j, c) = \max \{ H(i', j', c) : i' \in [i \cdot s, i \cdot s + p - 1], j' \in [j \cdot s, j \cdot s + p - 1] \} \quad (16)$$

where s is the stride. In this paper, we used $p = 2$ and $s = 2$. The final layer of the encoder is a fully connected layer.

Symmetrically, the decoder is composed of a fully connected layer as first layer, one convolutional layer, and one upsampling layer that repeats the rows and the columns of the input matrix by the same factor used during pooling, i.e., 2. The last layer of the network is composed of a convolutional layer with linear activation function, and one kernel so that the final output matrix has the same dimensions of \mathbf{X} .

As for the MLP autoencoder, the hyperbolic tangent function has been used as activation function and batch normalisation has been applied after the hidden layers. The depth of the network can be increased by adding more convolutional and fully connected layers to the encoder and consequently to the decoder, and the actual topology has been determined in the experiments by using a validation set.

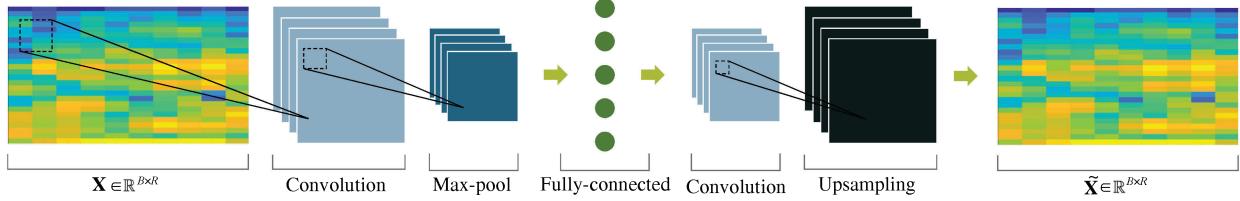


Fig. 2. Diagram of a convolutional neural network autoencoder with one convolutional and max-pooling layer, one fully-connected layer, and one convolutional and upsampling layers.

3) *Long Short-Term Memory (LSTM) Autoencoder*: Additionally to fully connected and convolutional autoencoders, we evaluated also networks composed of LSTM recurrent units (Fig. 3) [44]. These blocks can efficiently exploit a long-range temporal context by means of connections between units which form directed cycles, and store state information in the cell variable \mathbf{c}_r . As for the CNN autoencoder, the LSTM autoencoder operates on a sequence of R features vectors. Each cell consists of three gates: the input gate \mathbf{i}_r , the output gate \mathbf{o}_r , and the forget gate \mathbf{f}_r , as depicted in Fig. 3. The forget gate can reset the state variable which leads to “forgetting” the cell variable \mathbf{c}_r , while the input and output gates are responsible for reading input from \mathbf{x}_r and writing output to \mathbf{h}_r , respectively. The following equations define the behaviour of LSTM blocks:

$$\mathbf{z}_r = \tanh(\mathbf{W}_z \mathbf{x}_r + \mathbf{R}_z \mathbf{h}_{r-1} + \mathbf{b}_z) \quad (17)$$

$$\mathbf{i}_r = \sigma(\mathbf{W}_i \mathbf{x}_r + \mathbf{R}_i \mathbf{h}_{r-1} + \mathbf{b}_i) \quad (18)$$

$$\mathbf{f}_r = \sigma(\mathbf{W}_f \mathbf{x}_r + \mathbf{R}_f \mathbf{h}_{r-1} + \mathbf{b}_f) \quad (19)$$

$$\mathbf{c}_r = \mathbf{i}_r \otimes \mathbf{z}_r + \mathbf{f}_r \otimes \mathbf{c}_{r-1} \quad (20)$$

$$\mathbf{o}_r = \sigma(\mathbf{W}_o \mathbf{x}_r + \mathbf{R}_o \mathbf{h}_{r-1} + \mathbf{b}_o) \quad (21)$$

$$\mathbf{h}_r = \mathbf{o}_r \otimes \tanh(\mathbf{c}_r) \quad (22)$$

where \otimes denotes element-wise multiplication, and $\sigma(\cdot)$ and $\tanh(\cdot)$ denote respectively element-wise sigmoid and hyperbolic tangent functions. The matrices \mathbf{W}_z , \mathbf{W}_i , \mathbf{W}_f , and \mathbf{W}_o are rectangular matrices containing the input weights, while the matrices \mathbf{R}_z , \mathbf{R}_i , \mathbf{R}_f , and \mathbf{R}_o are square matrices that contain recurrent weights. Finally, \mathbf{b}_z , \mathbf{b}_i , \mathbf{b}_f , and \mathbf{b}_o are the bias vectors. At each time-step r , an LSTM unit calculates the value the output \mathbf{h}_r from the cell variable \mathbf{c}_r and the output gate \mathbf{o}_r . The cell variable \mathbf{c}_r is computed by using the input gate \mathbf{i}_r , the value of \mathbf{z}_r , the forget gate \mathbf{f}_r , and the cell variable value at the previous time-step \mathbf{c}_{r-1} . Both \mathbf{i}_r and \mathbf{z}_r are calculated from the current input \mathbf{x}_r and the output of the LSTM unit at the previous time-step \mathbf{h}_{r-1} but with a different set of parameters. The forget gate assumes values in the range $(0, 1)$ and sets a weight for \mathbf{c}_{r-1} . The output of the LSTM cell \mathbf{h}_r is finally calculated from the current value of the cell variable \mathbf{c}_r multiplied by \mathbf{o}_r . The output gate \mathbf{o}_r acts for the output of the LSTM cell similarly to the forget gate for the cell variable \mathbf{c}_r . For additional details, please refer to [44].

As for the fully connected and convolutional autoencoder, the final topology of the LSTM network has been determined experimentally (see Section V).

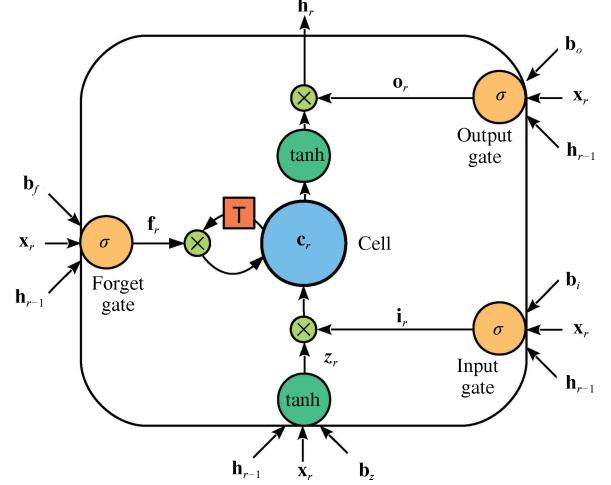


Fig. 3. A long short-term memory block, containing a memory cell and the input, output, and forget gates. The symbol “T” denotes a unitary time-lag.

C. Decision

A sequence is classified as faulty or non-faulty based on the magnitude of the reconstruction error. For a single feature vector, the error is calculated as follows:

$$e(r) = \|\tilde{\mathbf{x}}_r - \mathbf{x}_r\|. \quad (23)$$

Supposing that the entire sequence is composed of R feature vectors, the number of samples where $e(r)$ exceeds a certain threshold δ is defined as:

$$N_e = |\{e(r) : e(r) \geq \delta, r = 1, 2, \dots, R\}|. \quad (24)$$

The entire sequence, then, is classified as faulty only if N_e exceeds a predefined minimum number F .

This procedure has been adopted because although each sequence is associated with a single condition (faulty or non-faulty), it is unlikely that all frames in the sequence have the same error. During the signal recording phase, the robotic arm turns the motor and certain abnormal vibrations can be observed only for few time instants. For this reason, a sequence is considered faulty only if a minimum number of frames in the sequence exceeds the threshold. The performance for different values of F has been studied in the experiments.

IV. COMPARATIVE METHOD

The proposed approach has been compared with the OC-SVM algorithm [33]. A one-class SVM consists in a discriminant function that takes the value $+1$ in a small region that captures the majority of the data points of a set and -1 outside



Fig. 4. Diagram of the acquisition system. The vibration signal is acquired by using the PCB 352C33 accelerometer and is then amplified with the MMF M32 device. A National Instruments PCIe-6351 board is connected to the output of the amplifier and it is used to acquire the final signal.

that region [33]. The discriminant function has the following expression:

$$f(\mathbf{x}) = \text{sgn} \left(\sum_i \alpha_i \cdot k(\mathbf{x}_i, \mathbf{x}) - \rho \right), \quad (25)$$

where \mathbf{x}_i denotes the i -th support vector, and $k(\cdot, \cdot)$ represents the kernel function, e.g., the radial basis function $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$. The position of the hyperplane, thus, defines the region that represents normal data points. For each point \mathbf{x} that lies outside this region, the function $f(\mathbf{x})$ takes the value -1 , whereas for point inside the region, it takes the value $+1$.

The terms α_i can be found by solving the solution to the dual problem:

$$\min_{\alpha} \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j), \quad (26)$$

$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{\nu l}, \quad \sum_i \alpha_i = 1. \quad (27)$$

The term $\nu \in (0, 1]$ is an hyperparameter of the algorithm.

The offset ρ can be obtained from the Karush-Kuhn-Tucker (KKT) condition with the expression [45]:

$$\rho = \sum_j \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) \quad (28)$$

which is satisfied for any α_i that is not at the upper or lower bound.

The values of ν and γ have been determined in the experiments.

V. EXPERIMENTS

A. Data

1) *Signal Acquisition and Pre-Processing*: The signals used for this study derive from an actual industrial control system placed at the end of a DC motor production line of an automotive components manufacturer. The system is composed of a robotic arm that picks and moves the motor under test and a measuring station (see Figs. 4 and 5).

The standard procedure for motor quality check with a human operator consists in the following steps: taking a motor from a quality inspection bin; performing visual inspection of the motor and check if all components are assembled correctly; connecting the power cable; checking that the shaft rotates; rotating the motor in order to assess noise and vibration at different tilt angles; disconnecting the power cable; assigning a verdict (pass or fail). The weight of the rotor creates an unbalance force when the rotor axis is not aligned with gravity,

so it is important to test the motor in different tilt angles, paying particular attention to the angle at which the motor will be mounted in the final deployment.

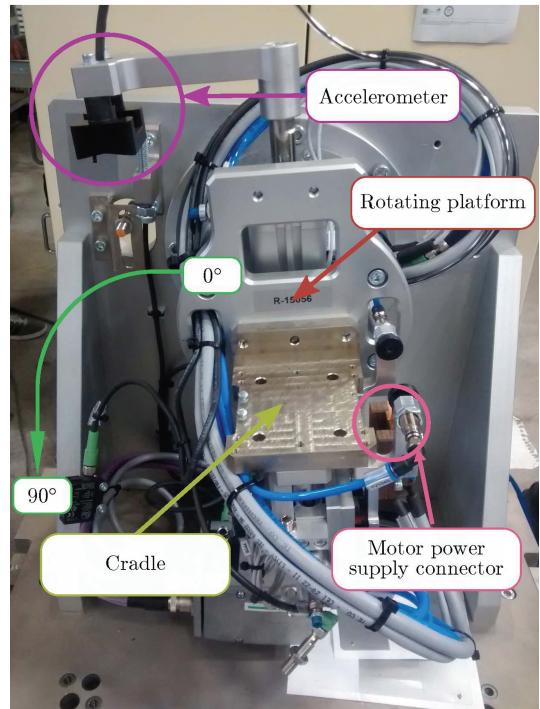


Fig. 5. Measuring bench for the end-line motors quality check.

The automatic control system mimics the operations of the human operator and performs the following steps (Fig. 5):

1) A robot arm takes the motor from the line and places it horizontally on the “cradle” the measuring station.

2) The clamp is closed, and it is lowered by rotating 90° to keep the engine in the cradle. In the upper part of the clamp there is a single-axis PCB 352C33 accelerometer (sensitivity of 100 mV/g) with IEPE technology. The pressure of the pneumatic cylinder that moves the clamp is monitored and kept constant in order to have a constant motor-craddle coupling force and thus increasing the repeatability of the measure.

3) During step number 2, the entire measurement area is released so that it remains floating on pneumatic suspension so as to avoid vibration transmission from the floor.

4) A 3 V power supply is applied with a TDK lambda ZUP 20-20 power supply driven by an analogue output of a National Instrument PCI express 6351 board. At this voltage, the motor accelerates up to 1000 rpm. A PID controller regulates the voltage so that its speed is $1000 \pm 5 \text{ rpm}$. It is therefore possible to assume that the tests take place at constant speed of

1000 rpm.

5) Once the speed is stable, the actual test in which vibrations are measured starts. During the test, the motor is rotated 90°, from the horizontal to the vertical position in 3 s. This rotation is necessary because some motors are noisy only horizontally, others only in vertical, depending on the defect.

6) After the test is concluded, the motor returns quickly horizontally, the clamp is opened, the robot takes it and puts it back on the line. In the case of a faulty motor, it is tested a second time, and if the fault is confirmed it is definitely classified as faulty.

The acceleration signals are acquired by using a National Instruments PCIe-6351 board with a sample rate of 25.6 kS/s (Fig. 4). The acquired signal is processed by the MMF Integrated Electronics Piezo Electric (IEPE) M32 signal conditioner, which is equipped with an anti-aliasing filter with cut-off frequency set at 10 kHz.

A segment of length 3 s is extracted from each acquired signal: this segment corresponds to the time interval during which the motor is kept at constant speed. An example of a vibration signal is shown in Fig. 6 both in the time and frequency domain. Observing Fig. 6(b), it is evident that the majority of the signal energy is located below 6 kHz. This motivated us to downsample the acquired signals at 12.8 kS/s in order to maintain only the portion of spectrum containing the majority of the information. Moreover, signals are filtered with a high-pass filter with cut-off frequency equal to 10 Hz.

B. Experimental Setup

The dataset used in the experiments contains 1178 signals associated with non-faulty motors and 14 signals that human operators have classified as defective. These data have been divided into a training set containing 1170 non-defective motor signals, and a test set containing 14 faulty signals and 8 non-faulty signals.

Regarding the feature extraction stage, as mentioned in Section III-A vibration signals are divided in frames 600 ms long (7680 samples) overlapped by 300 ms (3840 samples). For signals with bandwidths in the range 4–8 kHz, the number of filters of the filterbank is usually chosen empirically between 20 and 40 [35], [36]. The bandwidth of the acquired vibration signals is about 6 kHz, and preliminary experiments showed that using 26 filters provided a reasonable fault detection performance.

Several tests have been carried out to find the most performing topology of the MLP, CNN, and LSTM autoencoders. All the networks have been trained by using the Adam algorithm [46] with the learning rate $\alpha = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Additionally, the following noise variance values have been evaluated: $\sigma \in \{0.0, 0.01, 0.1, 0.25, 0.5\}$.

Different network topologies have been explored by using the random search strategy [47]. For each type of autoencoder, the number of explored topologies is 400, and details on the prior distributions used for generating the hyperparameters values are shown in Table I. Note that for the CNN autoencoder, only the number of convolutional layers of the encoder is indicated: as described in Section III-B-2, the

encoder comprises also one fully connected layer, and the overall network is composed of twice the number of layers of the encoder plus a final convolutional layer. Regarding the number of frames F for deciding whether a sequence contains a fault or not, its influence has been evaluated by varying the value from 2 to 8.

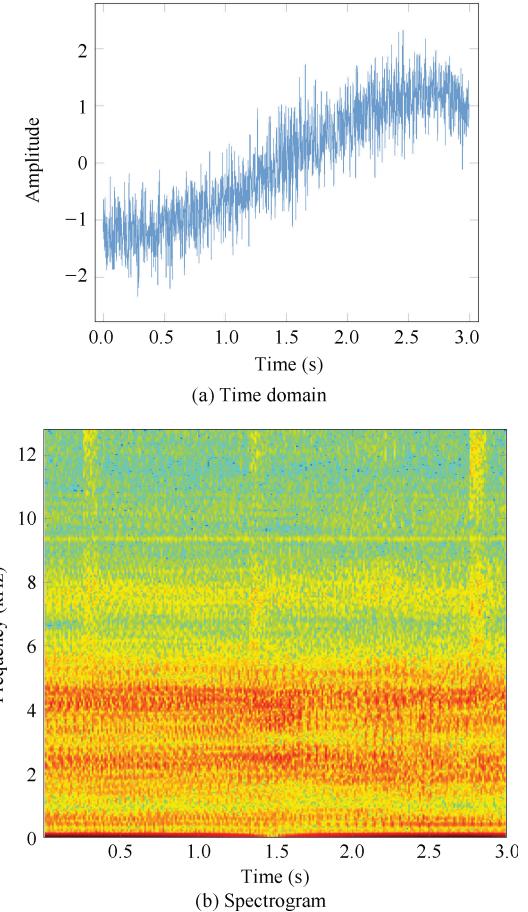


Fig. 6. An example of the acquired vibration signal.

TABLE I
PARAMETERS EXPLORED DURING THE RANDOM SEARCH ALONG WITH THEIR PRIOR DISTRIBUTIONS. $\log_2 U$ INDICATES A UNIFORM DISTRIBUTION IN THE LOG-DOMAIN

	Parameter	Prior
MLP	Nr. of layers	2, 3, or 4
	Nr. of hidden units	$\log_2 U(16, 512)$
CNN	Nr. of encoder layers	1, 2, or 3
	Nr. of kernels	$\log_2 U(4, 64)$
LSTM	Kernel size	3 or 5
	Nr. of layers	2, 3, or 4
	Nr. of hidden units	$\log_2 U(16, 512)$

The results obtained with the neural network have been compared with those obtained with OC-SVM. OC-SVM models have been trained with radial basis function kernel and different values of γ and ν parameters: $\gamma \in \{2^{-5}, 2^{-4}, \dots, 2^{-1}, 2^0\}$, and $\nu \in \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}\}$.

The performance has been evaluated by using the receiver operating curve (ROC) obtained by varying the decision threshold and the related area under curve (AUC).

C. Results

The results obtained for the different networks topologies and the OC-SVM algorithm are shown in Fig. 7. Observing the results, it is evident that the most performing algorithm is the MLP autoencoder, which achieved an AUC equal to 99.11 %. The OC-SVM is the lowest performing approach, with an AUC equal to 81.14 %. The LSTM autoencoder, on the other hand, performs similarly to the MLP, with an AUC equal to 98.21 %. The lowest performing of the network-based algorithms is the CNN autoencoder, with an AUC equal to 84.82 %. Table II shows the final topologies of the three autoencoders and the values of γ and ϵ for the OC-SVM algorithm.

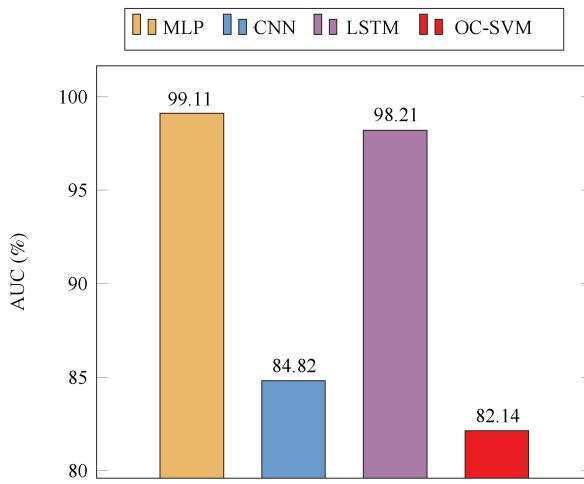


Fig. 7. Results obtained for the different autoencoder topologies and the OC-SVM algorithm.

TABLE II
HYPERPARAMETERS VALUES DETERMINED IN
THE RANDOM AND GRID SEARCH

	Parameter	Value
MLP	Nr. of layers	4
	Nr. of hidden units per layer	423, 171, 325, 26
	F	6
	σ	0
	Nr. of encoder layers	3
	Nr. of kernels per layer	12, 21, 10
CNN	Kernel size	5×5
	Nr. of units in fully connected layers	82, 80
	Kernel size of the last layer	7×7
	F	5
	σ	0.01
	Nr. of layers	2
LSTM	Nr. of LSTM units per layer	52, 52
	F	6
	σ	0
OC-SVM	γ	2^{-4}
	ν	2^{-1}
	F	2

Fig. 8 shows two example sequences related to a non-faulty (Fig. 8 (a)) and a faulty (Fig. 8 (b)) motor. More in details, the top figures (Fig. 8 (a) and Fig. 8 (b)) show the original signals, the middle figures (Fig. 8 (c) and Fig. 8 (d)) the signals reconstructed by the MLP network, and the bottom figures the errors used for the decision (Fig. 8 (e) and Fig. 8 (f)). Both observing the reconstructed signal and the error, it is evident that the network is able to reconstruct an input sequence related to a non-faulty motor with a considerably lower error than in the case of an input sequence related to a faulty motor. Indeed, in the first case the peak error value is equal to 2.03, while in the second case it is equal to 0.16.

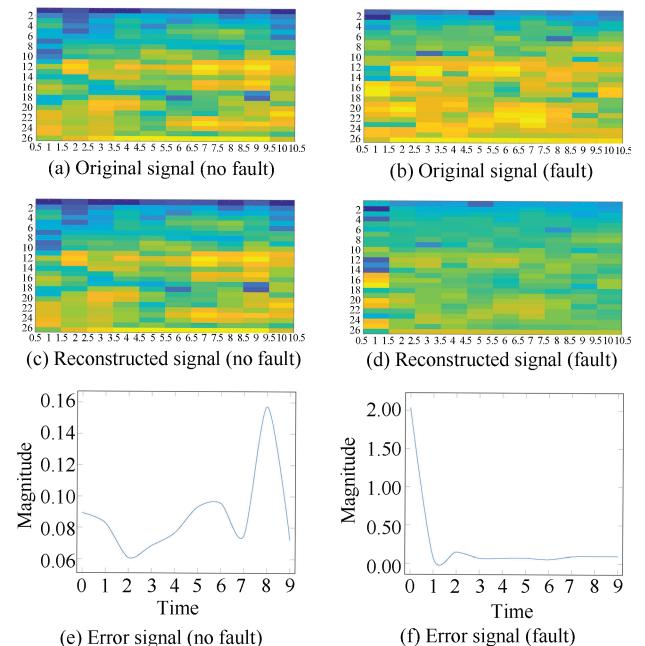


Fig. 8. On the left side, the original non-faulty signal (a) and the signal reconstructed by the MLP neural network (c), along with error signal (e). On the right side, the same signals related to an original signal of a faulty motor.

Fig. 9 shows the results obtained for different values of the standard deviation of the Gaussian noise added to the input of the autoencoders during the training phase. The results show that the optimal value of σ depends on the architecture of the autoencoders, however the highest AUCs have been obtained either without noise (MLP and CNN), or with a low value of σ (LSTM). This suggests that the autoencoder has good generalisation capabilities also without corrupting the input with the noise signal.

The results obtained by varying the minimum number of frames F necessary for classifying a sequence as faulty are shown in Fig. 10. Regardless the algorithm, the lowest value of AUC has been obtained with $F = 1$. This confirms what observed in Section III-C, i.e., that classifying an entire sequence based only on a single value of the error is not sufficient and can result in a high number of false positives. The three evaluated autoencoders exhibit a similar behaviour, with increasing values of AUC up to $F = 6$ and decreasing values for greater F . When $F = 1$, the MLP algorithm is the lowest performing approach: indeed, the recurrent behaviour

of LSTM autoencoder provides significant benefits over the memoryless MLP, providing a 16.07 % improvement over the latter autoencoder. Similarly, the CNN autoencoder processes the entire sequence of feature vectors, thus retaining all the contextual information of a sequence. With $F = 1$, this results in a 24.10 % absolute improvement compared to the MLP.

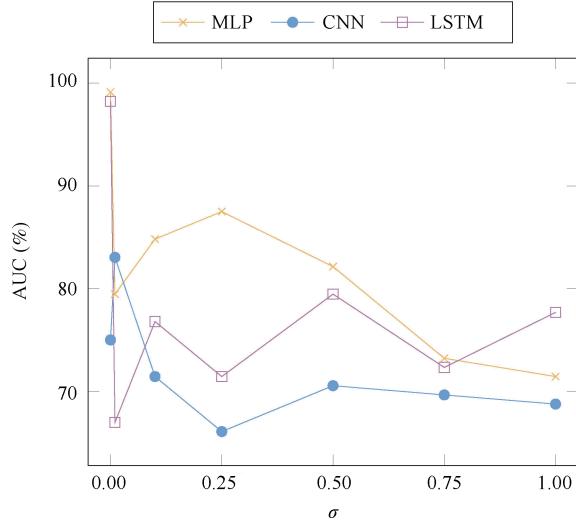


Fig. 9. Performance of the three evaluated autoencoders for different values of standard deviation of Gaussian noise signal.

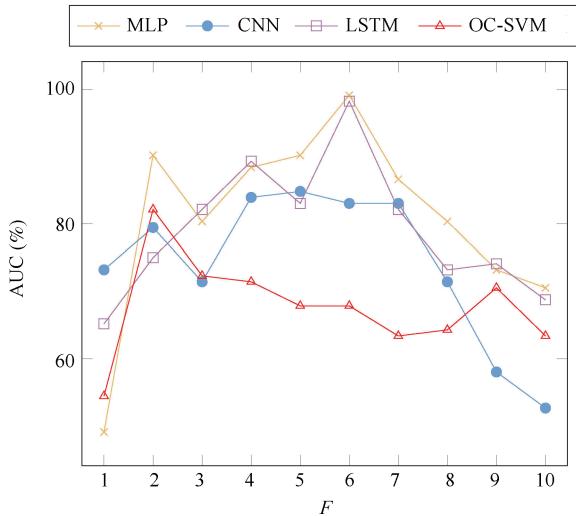


Fig. 10. Results obtained for the different autoencoder topologies and values of F .

D. Discussion

The results of the experiments evidenced that the all proposed autoencoder-based approaches outperform the OC-SVM approach. Comparing the performance of the MLP, LSTM, and CNN autoencoders (Fig. 7), it is also evident that the MLP autoencoder is the most performing among the evaluated topologies. Respect to the CNN autoencoder, the ROC-AUC difference is 14.29 %, while compared to the LSTM autoencoder, the difference is equal to 0.90 % and is more modest. In summary, for the case-study taken into consideration, the

MLP autoencoder is the most performing of the evaluated topologies.

VI. CONCLUSION

This paper presented an unsupervised method for detecting faults of electric motors based on vibration signals and deep neural networks. Vibration signals have been acquired from an actual industrial control system placed at the end of a DC motor production line of an automotive components manufacturer. The proposed method belongs to the family of novelty detection approaches, and, up to the authors' knowledge, this is the first time that they are used for detecting motor faults. The proposed solution consists in extracting Log-Mel coefficients from vibration signals and detecting faults by using autoencoders networks. Three different neural networks architectures have been evaluated: multi-layer perceptron autoencoder, convolutional neural network autoencoder, and long short-term memory autoencoder. The experimental evaluation has been conducted by using a dataset created by the authors by acquiring the vibration signals of DC electric motors. The signals have a sample rate equal to 25.6 kS/s and they have been acquired from an industrial control system placed at the end of the production line of an automotive components manufacturer. More than 1000 sequences have been used for training the algorithms, and 22 sequences for testing. The performance has been evaluated in terms of area under ROC curve, and the results showed that the MLP autoencoder outperforms the other architectures, as well as the comparative method based on one-class support vector machine.

Future developments will be oriented to evaluate different neural networks architectures, such as generative adversarial networks [18], [48], Siamese networks [49], [50], and non-intrusive approaches [51], [52]. Moreover, different input features and the influence of their hyperparameters will be studied and compared to end-to-end approaches [53], [54]. The proposed methodology will be applied to different types of electrical motors and generators, addressing a large variety of faults occurring in their standard operating conditions as well. The robustness of the algorithms will be evaluated also by varying the speed and the load of the motors under test [55].

ACKNOWLEDGEMENT

This research has been partly supported by the Italian University and Research Consortium CINECA. We acknowledge them for the availability of high performance computing resources and support.

REFERENCES

- [1] Y. Merizalde, L. Hernández-Callejo, and O. Duque-Perez, "State of the art and trends in the monitoring, detection and diagnosis of failures in electric induction motors," *Energies*, vol. 10, no. 7, 2017.
- [2] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.
- [3] Z. Gao, C. Cecati, and S. Ding, "A survey of fault diagnosis and faulttolerant techniques-part i: Fault diagnosis with model-based and signalbased approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

- [4] Z. W. Gao, C. Cecati, S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques-part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3768–3774, 2015.
- [5] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.
- [6] 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, 2005.
- [7] M. Seera, C. P. Lim, S. Nahavandi, and C. K. Loo, "Condition monitoring of induction motors: A review and an application of an ensemble of hybrid intelligent models," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4891–4903, 2014.
- [8] R. Isermann, "Model-based fault-detection and diagnosis—status and applications," *Annual Reviews in Control*, vol. 29, no. 1, pp. 71–85, 2005.
- [9] M. Bouzid and G. Champenois, "New expression of symmetrical components of the inductor motor under stator faults," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 4093–4410, 2013.
- [10] A. Adouni, A. Abid, and L. Sbita, "A DC motor fault detection, isolation and identification based on a new architecture Artificial Neural Network," in *Proc. 5th Int. Conf. on Systems and Control (ICSC)*, 2016, pp. 294–299.
- [11] P. Konar and P. Chattopadhyay, "Bearing fault detection of induction motor using wavelet and support vector machines (SVMs)," *Applied Soft Computing Journal*, vol. 11, no. 6, pp. 4203–4211, 2011.
- [12] V. N. Ghate and S. V. Dudul, "Cascade neural-network-based fault classifier for three-phase induction motor," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1555–1563, 2011.
- [13] R. Razavi-Far, M. Farajzadeh-Zanjani, S. Zare, M. Saif, and J. Zarei, "One-class classifiers for detecting faults in induction motors," in *Proc. 30th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017.
- [14] A. Soualhi, G. Clerc, and H. Razik, "Detection and diagnosis of faults in induction motor using an improved artificial ant clustering technique," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4053–4062, 2013.
- [15] H. C. Cho, J. Knowles, S. Fadali, and K. S. Lee, "Fault detection and isolation of induction motors using recurrent neural networks and dynamic Bayesian modeling," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 430–437, 2010.
- [16] M. Markou and S. Singh, "Novelty detection: A review—part 1: Statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [17] M. Markou and S. Singh, "Novelty detection: A review—part 2: Neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, 2003.
- [18] E. Principi, F. Vesperini, S. Squartini, and F. Piazza, "Acoustic novelty detection with adversarial autoencoders," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, Anchorage, AK, USA, 2017, pp. 3324–3330.
- [19] T. Schlegl, P. Seeböck, S. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10265 LNCS, pp. 146–147, 2017.
- [20] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers and Security*, vol. 28, no. 1–2, pp. 18–28, 2009.
- [21] C. Gong, "Exploring commonality and individuality for multi-modal curriculum learning," in *Proc. 31st AAAI Conference on Artificial Intelligence*, 2017, pp. 1926–1933.
- [22] C. Gong, D. Tao, S. Maybank, W. Liu, G. Kang, and J. Yang, "Multimodal curriculum learning for semi-supervised image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3249–3260, 2016.
- [23] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1452–1465, 2017.
- [24] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph laplacian for semisupervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2261–2274, 2015.
- [25] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72–73, pp. 303–315, 2016.
- [26] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.
- [27] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, 2017.
- [28] Y. LeCun. LeNet-5, convolutional neural networks [Online]. Available: <http://yann.lecun.com/exdb/lenet>, 2015.
- [29] S. Zgarni, H. Keskes, and A. Braham, "Nested SVDD in DAG SVM for induction motor condition monitoring," *Engineering Applications of Artificial Intelligence*, vol. 71, pp. 210–215, 2018.
- [30] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 1, pp. 185–195, 2018.
- [31] F. Jia, Y. Lei, L. Guo, J. Lin, and S. Xing, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 619–628, 2018.
- [32] H. Shao, H. Jiang, Y. Lin, and X. Li, "A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders," *Mechanical Systems and Signal Processing*, vol. 102, pp. 278–297, 2018.
- [33] B. Scholkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*, vol. 12. MIT Press, 2000, pp. 582–588.
- [34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. Cambridge, Massachussets, USA: MIT Press, 2016, ch. 14, pp. 502–524.
- [35] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 7398–7402.
- [36] A.-R. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 4273–4276.
- [37] Y. Kashimoto, M. Fujimoto, H. Suwa, Y. Arakawa, and K. Yasumoto, "Floor vibration type estimation with piezo sensor toward indoor positioning system," in *Proc. Int. Conf. on Indoor Positioning and Indoor Navigation*, Madrid, Spain, 2016, pp. 1–6.
- [38] F. Nelwamondo and T. Marwala, "Faults detection using gaussian mixture models, mel-frequency cepstral coefficients and kurtosis," in *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 1, Taipei, China, 2007, pp. 290–295.
- [39] A. Oppenheim and R. Schafer, *Discrete-time Signal Processing*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 2009.
- [40] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [41] D. O'Shaughnessy, *Speech Communications: Human and Machine*, 2nd ed. IEEE, 1999.
- [42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd International Conference on Machine Learning (ICML)*, vol. 1, 2015, pp. 448–456.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 2016.
- [44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- [45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [46] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [47] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [48] M. Kliger and S. Fleishman, “Novelty detection with gan,” *arXiv:1802.10560*, 2018.
- [49] D. Droghini, F. Vesperini, E. Principi, S. Squartini, and F. Piazza, “Few-shot siamese neural networks employing audio features for humanfall detection,” in *Proc. International Conference on Pattern Recognition and Artificial Intelligence*, Union, NJ, USA, 2018, pp. 63–69.
- [50] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *ICML Deep Learning Workshop*, vol. 2, 2015.
- [51] U. Orji, Z. Remscrim, C. Laughman, S. Leeb, W. Wichakool, C. Schantz, R. Cox, J. Paris, J. Kirtley Jr., and L. Norford, “Fault detection and diagnostics for non-intrusive monitoring using motor harmonics,” in *Proc. IEEE Applied Power Electronics Conference and Exposition (APEC)*, Palm Springs, CA, USA, 2010, pp. 1547–1554.
- [52] R. Bonfigli, E. Principi, M. Fagiani, M. Severini, S. Squartini, and F. Piazza, “Non-intrusive load monitoring by using active and reactive power in additive factorial hidden markov models,” *Applied Energy*, vol. 208, pp. 1590–1607, 2017.
- [53] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proc. International Conference on Machine Learning (ICML)*, vol. 5, 2014, pp. 3771–3779.
- [54] W. Caesarendra and T. Tjahjowidodo, “A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing,” *Machines*, vol. 5, no. 4, 2017.
- [55] Z. Li, Y. Jiang, Q. Guo, C. Hu, and Z. Peng, “Multi-dimensional variational mode decomposition for bearing-crack detection in wind turbines with large driving-speed variations,” *Renewable Energy*, vol. 116, pp. 55–73, 2018.



Damiano Rossetti received the M.Sc. degree in electronic engineering in 2014 from Università Politecnica delle Marche (Italy). In 2014, he started his Ph.D. period at the Department of Information Engineering of Università Politecnica delle Marche in collaboration with the Research for Innovation Department of Loccioni Group. In 2016, he was at the University of Lincoln as visiting student for 6 months, where he carried on his research on machine learning algorithms for industrial applications. He received the Ph.D. degree in electronic engineering

in 2018. He is currently a Software Designer at the Loccioni Group.



Stefano Squartini (SM’12) got the Italian Laurea with honours in electronic engineering from University of Ancona (now Polytechnic University of Marche, UnivPM), Italy, in 2002. He obtained the Ph.D. degree at the same university (November 2005). He worked also as post-doctoral researcher at UnivPM from June 2006 to November 2007, when he joined the DII (Department of Information Engineering) as Assistant Professor in Circuit Theory. He is now Associate Professor at UnivPM since November 2014. His current research interests are in the area of computational intelligence and digital signal processing, with special focus on speech/audio/music processing and energy management. He is author and co-author of more than 190 international scientific peer-reviewed articles. He is Associate Editor of the *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Cybernetics*, and *IEEE Transactions on Emerging Topics in Computational Intelligence*, and also member of Cognitive Computation, Big Data Analytics and Artificial Intelligence Reviews Editorial Boards. He joined the Organizing and the Technical Program Committees of more than 70 International Conferences and Workshops in the recent past. He is the Organizing Chair of the IEEE CIS Task Force on Computational Audio Processing.



Emanuele Principi got the Italian Laurea with honors in electronic engineering from University of Ancona (now Università Politecnica delle Marche), Italy, in 2004. He obtained the Ph.D. degree at the same university in November 2009. He is now a Post-doctoral Researcher since March 2010. His current research interests are in the area of digital signal processing and computational intelligence, with special focus on energy management and speech/audio processing. Dr. Principi has actively participated to various (funded) regional, national and European projects on multimedia Digital Signal Processing. He is author and co-author of many international scientific peer-reviewed articles, and he has been serving as a reviewer for several international journals, and conference proceedings. He is member of the editorial board of *Neural Computing and Applications* and *Artificial Intelligence Review* both edited by Springer from 2017, and member of the program committee of several international conferences. He also served as Guest Editor for the Special Issue on “Theory and Application of Computational Intelligence in Electric Vehicles and their Integration within Smart Energy Networks” (*Energies*, MDPI) and he organized the Special Session on “Deep Neural Audio processing” within the IEEE International Joint Conference on Neural Networks in 2017, 2018, and 2019. He is also member of the Texas Instrument Expert Advisory Panel and of the International Neural Networks Society.



Francesco Piazza got the Italian Laurea with honours in electronic engineering from the University of Ancona, Italy, in 1981. From 1981 to 1983 he worked on image processing at the Physics Department. In 1983 he worked at the Olivetti OSAI Software Development Centre (Ivrea, Italy). In 1985 he joined the Department of Electronics and Automatics of the University of Ancona. Currently he is Full Professor of Electrical Science and Director of the Department of Information Engineering at the Università Politecnica delle Marche, Italy. He is author or co-author of more than 350 international papers. His current research interests are in the areas of circuit theory and digital signal processing including adaptive DSP algorithms and circuits, artificial neural networks, speech and audio processing.