A Benchmark for Fault Detection in Evolving Environments and a Method of Solution Based on Sparse Autoencoder-Based Deep Neural Networks

Zhe Yang[1], Piero Baraldi[1*], Enrico Zio[1,2,3,4]
piero.baraldi@polimi.it
[1] *Energy Department, Politecnico di Milano, Via La Masa 34, 20156 Milan, Italy*
[2] *MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France*
[3] *Aramis Srl, Via pergolesi 5, Milano, Italy*
[4] *Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea*

Abstract. Degradation, maintenance, renewal and operational mode changes continuously modify the conditions of industrial systems. The identification of the onset of abnormal conditions from signal measurements taken in such evolving environments can be quite challenging, due to the difficulty of distinguishing the real cause of the signal variations. In this work, we address the problem of fault detection in evolving environments. 1) We propose a synthetic case study, which can serve as benchmark for future methods; 2) we present a method which uses a Sparse Autoencoder-based Deep Neural Network (SAE-DNN) within a novel procedure that remarkably reduces the computational burden for setting the values of the hyperparameters. The method is applied also to a bearing vibration dataset. The results show that it is able to accurately detect faults in evolving environments, outperforming other state-of-the-art methods.

Keywords. Deep learning, Sparse autoencoder, Deep neural network, Fault detection, Evolving Enrivonment

1. Introduction

Digitalization is one of the pillars of Industry 4.0. Under this paradigm, signals collected by sensors are elaborated to detect, diagnose, predict anomalous behaviors of components and systems.

In this work, we are concerned with fault detection, to avoid unplanned system breakdowns and severe consequences [1, 2]. Approaches, such as auto-associative kernel regression [3-5], fuzzy similarity [6, 7], k-nearest neighbor [8, 9], naive bayes classifier [10], decision tree [11], support vector machine [12], Artificial Neural Network (ANN) [13] and Extreme Learning Machine (ELM)

[14, 15] are typically used to elaborate the sensor data for discriminating faulty conditions. Some difficulties of current fault detection methods are: 1) they require a periodic retraining of the empirical model established on the training set initially available, because of the evolving environment in which the components and system operate, due to maintenance interventions, upgrading technological and operational plans, equipment degradation which modify the operational state of the monitored system, and, therefore, deteriorate the fault detection performance of the trained model, 2) they rely on the identification of features handcrafted resorting to expert knowledge or found by computationally demanding feature selection and extraction methods [16-19].

To contribute to addressing these difficulties, in this work we consider deep-learning techniques. These techniques are capable of extracting hierarchical representations of large-dimensional datasets by using neural networks with multiple layers of non-linear transformations [20] and have been successfully applied to various areas such as computer vision [21, 22], automatic speech recognition [23-25] and natural language processing [26, 27]. One of the most popular deep learning architecture is the AutoEncoder (AE) [28], which is a neural network with symmetrical structure, composed of an "encoder" and a "decoder" network. The multilayer "encoder" network transforms the large-dimensional input data into a small number of features and the "decoder" network recovers the data from the extracted features. An AE can be transformed into an AE-based DNN (AE-DNN) by adding a classification/regression layer on top of the encoder network. AE-DNN training is facilitated by the availability of an already pre-trained AE, which is, then, fine-tuned using input-output data.

Recently, DNNs based on AE and its variants, such as Sparse AutoEncoders (SAEs) [29] and Denoising AutoEncoders (DAEs) [30], have been applied to fault diagnostic problems [31-35]. An AE-DNN which receives in input frequency spectra is proposed for hydraulic pump fault diagnostics in [31]. A SAE-DNN is developed for gear pitting fault detection in [32]. An AE-DNN fed by frequency spectra of vibration data is used for rotating machinery fault diagnostics in [33]. A SAE is used to extract representative features from statistical metrics of bearing signals and a deep belief network is used to identify the bearing health condition in [34]. A method based on SAE-DNN for fault detection and diagnostics of combustion engines using acoustic signals is proposed in [35]. A DAE-DNN is developed for fault diagnosis of rotary machinery components in [36].

Even though the above-mentioned studies have outperformed other state-of-the-art fault detection and diagnostic methods, their use is complicated by the necessity of setting DNN hyperparameters, which is typically performed by trial-and-error and, therefore, can be computationally very intensive [37]. Furthermore, the approaches proposed have not considered the typical evolving environment setting of real industrial applications [38].

In this context, the objective of the present work is to develop a fault detection method with the following characteristics: 1) it works in evolving environments, 2) it does not require the application of feature selection and extraction techniques, 3) it does not require the use of time-consuming trial-and-error approaches for hyperparameters setting.

We propose the use of SAE-DNNs given that: *a*) the high-level data representation provided by the autoencoders allows obtaining a fault detection model robust to evolving environments; *b*) the SAE-DNN are able to work in large dimensional input signal spaces, without the necessity of using any feature extraction and selection approaches. From the methodological point of view, a novel procedure for setting the model hyperparameters during the SAEs pretraining phase, which is based on the use of a single run-to-failure degradation trajectory, is proposed. The idea is that, being the degradation process irreversible, a satisfactory high-level representation of a run-to-failure trajectory allows extracting a monotonic feature representing component degradation. In practice, good architectures and hyperparameter setting transform the high dimensional input signal space into a health indicator by just using SAEs, without any supervised training. Therefore, different hyperparameter settings are compared considering the monotonicity of features extracted from the SAEs, and the one which provides the most monotonic feature is selected. Notice that the computational efforts necessary for: 1) pre-training the model using large datasets with several run-to-failure trajectories 2) fine-tuning the model to set hyperparameters by trial-and-error are saved. Once the SAE hyperparameters are set, monitoring data collected in evolving environments are used to fine-tune the SAE-DNN for preventing the reduction of its performance in evolving environments.

The datasets available for the verification of fault detection methods can be classified into those obtained by simulating the operation of industrial equipment using physics-based models and those containing real signal measurements. The former class includes datasets simulating the operation of a ship propulsion system [39], a valve actuator system [40], an electro-mechanical position servo system [41] and a wind turbine system [42]. Since the fault scenarios are artificially created, the time of the onset of the abnormal conditions is known and, therefore, the performance of the fault detection methods can be precisely estimated. However, differently from real-world situations, they typically contain a small number of signals and they are not able to realistically reproduce the evolving environments. The latter class includes datasets containing real signal measurements collected from bearings [43], gearboxes [44], anemometers [45], power plants [46] and vehicles [47, 48]. These datasets contain various types of faults of a specific component (system) in different working conditions. Their use for the verification of fault detection methods is, however, complicated by the fact that the true time of the onset of the abnormal conditions is typically not known. Also, several of these datasets focus on a single component (system) and do not consider one of the most common causes of evolving environments, which is the effects on the monitored component (system) of the degradation and failure of other components (systems).

Given the limitations of the available datasets for the validation of fault detection methods, in this work we present a novel synthetic cases study in which the presence of evolving environment is simulated considering the effects of the degradation and failures of other components on the monitored one whose normal/abnormal state is known. The effectiveness of SAE-DNN pre-training using a single run-to-failure trajectory is investigated on the synthetic dataset and the performance of the developed SAE-DNN model is evaluated in terms of missed and false detection rates, and compared to that of commonly used fault detection methods. Notice that the proposed synthetic dataset can serve as benchmark for the validation of future fault detection methods in evolving environment.

The proposed method is further validated by a benchmark case study of bearing degradation [43] in an evolving environment.

The remaining of this paper is organized as follows. The basic theory of SAE-DNN is introduced in Section 2. Section 3 presents the problem statement. In Section 4, the proposed fault detection method is illustrated. Section 5 presents the synthetic case study designed for validating the proposed fault detection method in evolving environments. Section 6 shows the applications of the proposed method to a bearing degradation case study. Finally, conclusions are drawn in Section 7.

2. Sparse Autoencoder-based Deep Neural Networks

This Section briefly illustrates the basic principles of SAE-DNNs. For generality, we consider the generic problem of building an empirical mapping function from a K-dimensional input signal space

$\mathbf{X} \subset \mathbb{R}^K$ to an output space $\mathrm{Y} \subset \mathbb{R}$ using N input-output training patterns $(\boldsymbol{x}_i, y_i)_{i=1,\ldots,N} \in \mathbf{X} \times \mathrm{Y}$.

2.1. Sparse autoencoders

An AE is a neural network composed by an encoder and a decoder network [28] (Figure 1**).**

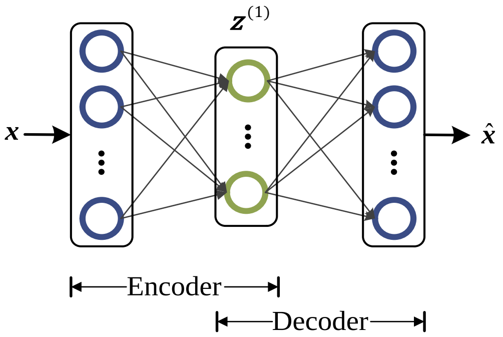

Figure 1. Structure of a basic AE

The encoder transforms an input vector $\boldsymbol{x} \in \mathbf{X}$ to its hidden representation $\boldsymbol{z}^{(1)} = \left[z_1^{(1)}, z_2^{(1)}, \ldots, z_{K_1}^{(1)}\right]$:

$$\boldsymbol{z}^{(1)} = f(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) \tag{1}$$

where f, $\boldsymbol{W}_1$, $\boldsymbol{b}_1$ are the encoder activation function, the weight matrix and the bias vector, respectively. Then, the decoder maps the encoded feature vector $\boldsymbol{z}^{(1)}$ back into the reconstruction $\hat{\boldsymbol{x}}$ of the input vector $\boldsymbol{x}$:

$$\hat{\boldsymbol{x}} = g\left(\boldsymbol{W}_2\boldsymbol{z}^{(1)} + \boldsymbol{b}_2\right) \tag{2}$$

where g, $\boldsymbol{W}_2$, $\boldsymbol{b}_2$ are the decoder activation function, the weight matrix and the bias vector, respectively. A SAE is a variant of an AE, which encourages the extraction of discriminative features [49]. SAE training, which uses the available input patterns $\boldsymbol{x}_i$, $i = 1, \ldots, N$, aims at minimizing the cost function:

$$f_{cost} = R_{error} + \beta R_{sparse} + \lambda R_{L2} \tag{3}$$

where R_{error}, R_{sparse} and R_{L2} are the reconstruction error, the sparsity regularization and the L_2 regularization terms, respectively, and β and λ are coefficients indicating the relative importance of the terms in the cost function. The reconstruction error, R_{error}, quantifies the dissimilarity between the reconstruction $\hat{\boldsymbol{x}}_i$ provided by the decoder, and the input training vector $\boldsymbol{x}_i$:

$$R_{error} = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 \tag{4}$$

It has been shown that the extraction of discriminative features $\boldsymbol{z}^{(1)}$ is favored when the hidden neurons are constrained to be inactive most of the times, i.e. by requiring the sparsity of the AE [29].

To this aim, a sparse regularization term, R_{sparse}, is added to the cost function of SAE (Eq. (3)). Let $\hat{p}_j$ be the average activation of the j-th hidden neuron of the SAE hidden layer, evaluated over all the training patterns $\boldsymbol{x}_i \in \mathbf{X}, i = 1, \cdots, N$:

$$\hat{p}_j = \frac{1}{N} \sum_{i=1}^{N} z_{i,j}^{(1)} \tag{5}$$

where $z_{i,j}^{(1)}$ is the j-th element of the i-th hidden representation $\boldsymbol{z}_i^{(1)}$, $j = 1, \cdots, K_1$. Since the sparsity condition requires that the activation of the hidden neurons is mostly zero, all SAEs hidden neurons should be characterized by a small value of $\hat{p}_j$, e.g. $\hat{p}_j = p = 0.05$, $j = 1, \cdots, K_1$. R_{sparse} is, then, calculated using the Kullback-Leibler (KL) divergence function to measure the sparsity conditions, i.e. whether $\hat{p}_j$ is close to a desired sparsity proportion p:

$$R_{sparse} = \sum_{j=1}^{K_1} \mathrm{KL}(p \| \hat{p}_j) = \sum_{j=1}^{K_1} \left[p \log \frac{p}{\hat{p}_j} + (1 - p) \log \frac{1-p}{1-\hat{p}_j} \right] \tag{6}$$

Therefore, the KL function is zero when all $\hat{p}_j$ are equal to p and increases when they diverge.

The $L2$ regularization term, R_{L2}, which is introduced in Eq. (3) to prevent that R_{sparse} becomes small during the training process due to the increase of the weight values [49], is defined by:

$$R_{L2} = \frac{1}{2} \|\boldsymbol{W}\|^2 \tag{7}$$

where $\boldsymbol{W}$ is the SAE weight matrix.

2.2. Deep neural networks

DNNs learn the mapping from the high dimensional input signal space to the output space by automatically extracting high-level representations from input data using multiple layers of non-linear transformations. The construction of a SAE-DNN requires the training of multiple basic SAEs, which are then stacked to build multiple layers of the SAE-DNN. More specifically, the construction of a L-hidden-layer SAE-DNN requires (Figure 2):

i) a pre-training phase based on the sequential training of L basic SAEs (Figure 2(a)). Once the first basic SAE is trained and the features $\boldsymbol{z}_i^{(1)}$ have been extracted (Section 2.1), they are used as training "dataset" for building the next basic SAE, which transforms $\boldsymbol{z}_i^{(1)}$ to the feature vector $\boldsymbol{z}_i^{(2)}$. The procedure is repeated until the last basic SAE is trained. Then, all the basic SAEs are "stacked" to create a stacked SAE (Figure 2 (b)) [50].

ii) a fine-tuning phase. The stacked SAE is transformed into a SAE-DNN by adding a regression/classification layer on top of the encoder of stacked SAE (Figure 2 (c)). The obtained SAE-DNN is trained using the input data $\{\boldsymbol{x}_i, i = 1, \cdots, N\}$ and the corresponding output data $\{y_i, i = 1, \cdots, N\}$.

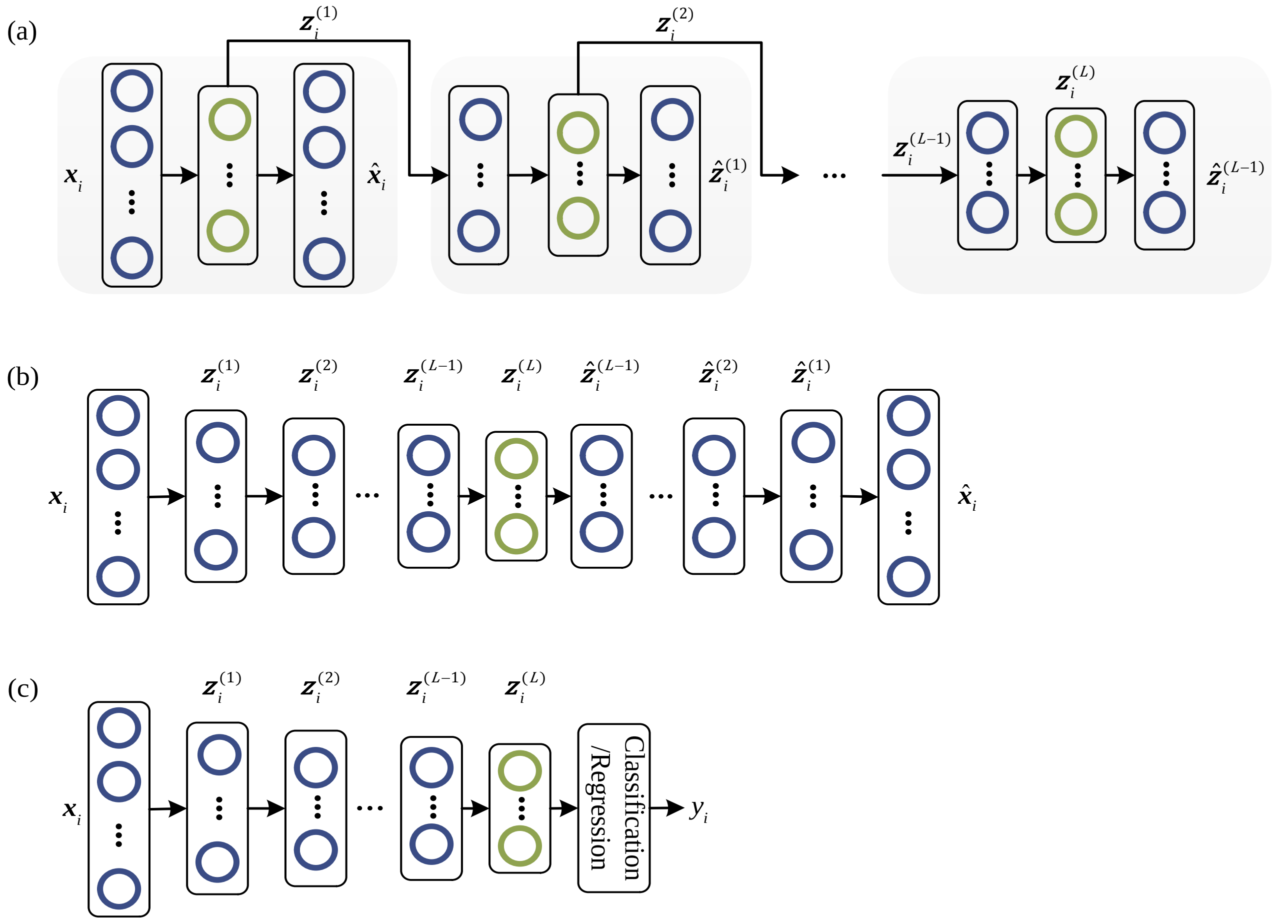

Figure 2. The procedure for building a L-hidden-layer deep neural network: (a) sequential training of L basic SAEs; (b) stacking of the the basic SAEs; (c) Transformation of the stacked SAE into a SAE-DNN

3. Problem Statement

The objective of the present work is to develop a fault detection method able to promptly identify the onset of abnormal conditions in industrial equipment operating in evolving environments. We assume to have available the measurements of K signals collected during the operation of similar equipment, and to know the normal/abnormal equipment state, $\mathbf{Y}$, at the time in which the measurements are collected. Let $n^{RTF} \geq 1$ be the number of observed run-to-failure trajectories, N_q the number of consecutive observations from the generic q-th trajectory, $q = 1, \ldots, n^{RTF}$, $\boldsymbol{x}_i^{RTF,q}$ the i-th vector of observations collected from the q-th trajectory and $y_i^{RTF,q}$ the corresponding equipment state with 0 indicating normal condition and 1 indicating abnormal condition. Matrixes $\mathbf{X}^{RTF} = \{\boldsymbol{x}_i^{RTF,q}, q = 1, \dots, n^{RTF}, i = 1, \dots, N_q\}$ and $\mathbf{Y}^{RTF} = \{y_i^{RTF,q}, q = 1, \dots, n^{RTF}, i = 1, \dots, N_q\}$ contain all the measurements and corresponding equipment states collected from run-to-failure trajectories. Measurements collected from incomplete degradation trajectories and the corresponding normal/abnormal states are organized into the matrixes $\mathbf{X}^{IN}$ and $\mathbf{Y}^{IN}$. All the available measurements and corresponding equipment states are concatenated in the matrixes $\mathbf{X}^{TRAIN} = \{\mathbf{X}^{RTF}; \mathbf{X}^{IN}\}$ and $\mathbf{Y}^{TRAIN} = \{\mathbf{Y}^{RTF}; \mathbf{Y}^{IN}\}$.

The fault detection method receives in input the test vector $\boldsymbol{x}^{TEST}$, containing the measurements collected at the present time, and is required to identify whether the component is in normal or abnormal condition, i.e. y^{TEST}= 0 or 1, respectively.

4. The proposed Fault Detection Method

The development of the proposed fault detection method is based on the two steps of: 1) building a stacked SAE using data collected from a single run-to-failure trajectory; 2) building the corresponding SAE-DNN by adding a softmax layer (classification layer) on top of the encoder of the stacked SAE and using data collected from several components working in different operating conditions for its training.

The stacked SAE architecture, i.e. the dimensions K_1, K_2,…, K_L of the hidden representation vectors $\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)}, \dots, \boldsymbol{z}_i^{(L)}$, is set considering the following principles:

i) the dimension of $\boldsymbol{z}_i^{(1)}$, K_1, is typically set larger than that of the input layer, K, to encourage the sparse decomposition of the input signals [49]. As shown in [49, 51-53], sparse-overcomplete representations have a number of theoretical and practical advantages, such as good robustness to noise and ability of extracting independent basis functions for input representation;

ii) the dimension K_l of the remaining hidden layers $\boldsymbol{z}_i^{(l)}$, $l = 2,3, \dots, L$, should be smaller than K_{l-1}, to encourage data compression;

iii) a sigmoid function is typically used as activation function [29].

A novelty of this work is the procedure for setting the SAE-DNN hyperparameters p, β and λ used in Eqs. (3) and (6). A stacked SAE is at first trained using a single run-to-failure trajectory and setting the hyperparameters equal to the default values $\bar{p} = 0.1$, $\bar{\beta} = 1$ and $\bar{\lambda} = 1e - 3$. If the number of available complete run-to-failure trajectories, n^{RTF}, is greater than 1, the trajectory to be used for the stacked SAE training is randomly sampled. The monotonicity of the K_L features $\boldsymbol{z}^{(L)}(j), j = 1, \dots, K_L$, extracted from the L-th inner layer of the stacked SAE are evaluated performing the Mann-Kendall (M-K) monotonicity test [54] and computing $|\rho_{MK}|$, i.e. the absolute value of the M-K coefficient ρ_{MK}. The idea is that a satisfactory high-level representation of the degradation process, which is intrinsically irreversible, should be provided by a monotonic feature. The M-K test is employed given its ability to assess the monotonic upward or downward trend of a signal over time, without requiring the knowledge of the signal distribution. It has been used to detect monotonic trends in series of environmental, climate and hydrological data [54]. The M-K coefficient of the generic j-th feature $\boldsymbol{z}^{(L)}(j) = \left\{z_{i,j}^{(L)}, i = 1, \dots, N_q\right\}$, extracted from the L-th hidden layer of the stacked SAE, is defined by:

$$\rho_{MK}\left(\boldsymbol{z}^{(L)}(j)\right) = \begin{cases} \frac{S-1}{\sigma_{MK}} & if\ \ S > 0 \\ 0 & if\ \ S = 0 \\ \frac{S+1}{\sigma_{MK}} & if\ \ S < 0 \end{cases} \tag{8}$$

where $S = \sum_{k=1}^{N_q-1} \sum_{i=k+1}^{N_q} \operatorname{sgn}\left(z_{i,j}^{(L)} - z_{k,j}^{(L)}\right)$, $\sigma_{MK} = \sqrt{N_q(N_q - 1)(2N_q + 5)/18}$ and $\operatorname{sgn}(\cdot)$ is the sign function. The larger the coefficient $\left|\rho_{MK}\left(\boldsymbol{z}^{(L)}(j)\right)\right|$, the more monotonic the feature $\boldsymbol{z}^{(L)}(j)$. The stacked SAE parameters, p, β and λ, are considered satisfactory if the most monotonic feature extracted from the L-th hidden layer, $\boldsymbol{z}^{(L)}(\tilde{j})$, $\tilde{j} \in [1, \dots, K_L\ ]$, has $\rho_{SAE}(p, \beta, \lambda) = \left|\rho_{MK}\left(\boldsymbol{z}^{(L)}(\tilde{j})\right)\right|$ larger than a threshold Th_{MK} set according to the procedure reported in appendix A. If $\rho_{SAE}(p, \beta, \lambda) < Th_{MK}$, new values of the stacked SAE hyperparameters are sampled and the corresponding stacked SAE built, until a stacked SAE with $\rho_{SAE}(p, \beta, \lambda) \geq Th_{MK}$ is found. Since p is the most sensitive among the three hyperparameters, we firstly tune it by randomly sampling its value in the range $[0, 0.5]$ with fixed $\bar{\beta} = 1$ and $\bar{\lambda} = 1e - 3$. If after $N_{max} = 20$ sampling of p, a set of hyperparameters with $\rho_{SAE}\left(p, \bar{\beta}, \bar{\lambda}\right) \geq Th_{MK}$ is not found, the second most sensitive hyperparameter of the model, λ, is varied. In practice, p is set to the value p_{best} at which $\rho_{SAE}\left(p, \bar{\beta}, \bar{\lambda}\right)$ has its maximum and $\log_{10} \lambda$ is randomly sampled from a discrete uniform distribution $\{-20, -19, \dots, 0\}$. Similarly, after $N_{max} = 20$ sampling of λ if $\rho_{SAE}\left(p_{best}, \bar{\beta}, \lambda\right) \geq$

Th_{MK} is not achieved, β is randomly sampled in [0, 10], with p_{best} and λ_{best} fixed at the values at which the most monotonic feature has been extracted.

With respect to step 2), a softmax layer with two neurons, representing normal and abnormal condition, respectively, is added on top of the stacked SAE to build a SAE-DNN. The training set is formed by all the available measurements, $\mathbf{X}^{TRAIN}$, and the corresponding 0/1 labels of the equipment normal/abnormal state, $\mathbf{Y}^{TRAIN}$. Each label y_i, $i = 1, \dots, N$, of $\mathbf{Y}^{TRAIN}$ is, then, transformed into a two dimensional vector $\left(y_{i,1}, y_{i,2}\right)_{i=1,\dots,N}$, which is associated to the two softmax neurons according to the following rule:

$$y_{i,1} = 1, y_{i,2} = 0 \ \text{ when } \ y_i = 0 \ \text{ (normal condition)}$$

$$y_{i,1} = 0, y_{i,2} = 1 \ \text{ when } \ y_i = 1 \ \text{ (abnormal condition)} \tag{9}$$

The developed SAE-DNN receives in input the test vector $\boldsymbol{x}^{TEST}$ containing the measurements collected at the present time and gives in output $\left(\hat{y}_1{}^{TEST}, \hat{y}_2{}^{TEST}\right)$, where $\hat{y}_1{}^{TEST} + \hat{y}_2{}^{TEST} = 1$. $\hat{y}_1{}^{TEST}$ is typically interpreted as an indicator proportional to the conditional probability that the equipment is in normal condition given $\boldsymbol{x}^{TEST}$ [55], i.e. $\mathrm{P}(\, y^{TEST} = 0 | \boldsymbol{x}^{TEST})$, and, dually, $\hat{y}_2{}^{TEST}$ is proportional to $\mathrm{P}(\, y^{TEST} = 1 | \boldsymbol{x}^{TEST})$. Either one of $\hat{y}_1{}^{TEST}$ and $\hat{y}_2{}^{TEST}$ is enough to define the equipment condition and in this work we focus on $\hat{y}_2{}^{TEST}$ to detect the occurrence of an abnormal condition when the value of $\hat{y}_2{}^{TEST}$ exceeds a pre-defined detection threshold $Th_d \in (0,1)$.

5. Synthetic case study

We apply the proposed method to a synthetic case study properly designed to reproduce the challenge of fault detection in evolving environments. The obtained results are compared with those obtained by ANN-based and ELM-based fault detection methods.

5.1. Data description

We consider 5000 similar machines operating in 10 plants in different locations and characterized by different operating conditions. The machine is formed by 4 identical components in parallel and its mission time T_d is equal to 1080 arbitrary time units (atu). The individual component degradation process is simulated by resorting to a degradation indicator $D(t)$, measured in arbitrary degradation units (adu), whose time evolution is:

$$D(t) = 0,\ t \leq \tau$$

$$D(t) = D(t-1) + \alpha\big(1 + 0.3 \cdot E(t-1) + w(t-1)\big),\ \ t > \tau \tag{10}$$

where $t = 1,2,\dots,T_d$, is the discretized time of evolution up to the mission time $T_d = 1080$ atu, τ is the time at which there is the onset of the degradation process and is sampled from an uniform distribution in the range $[650{,}750]$ atu, α is the degradation coefficient sampled from a gaussian probability density function with mean $\mu_\alpha = 0.015$ adu and standard deviation $\sigma_\alpha = 0.007$ adu, $E(t)$ is an adimensional "evolving environment" feature representing the machine operating condition at time t and $w(t) \sim N(0,\ 0.1)$ is the adimensional process noise.

A component enters in a critical functional state, i.e. the abnormal condition that we want to detect, when $D(t)$ exceeds the threshold $Th = 5$ adu. The abnormal condition of a component causes a modification of the operating conditions experienced by the other components of the same machine, which becomes harsher and, therefore, causes an acceleration of their degradation processes. This is modeled by assuming that the feature $E(t)$ representing the machine operating condition is:

$$E(t) = T(t) + 0.3N_f(t) \tag{11}$$

where $T(t)$ is an adimensional "temperature" feature proportional to the temperature experienced by the machine at time t, which depends from the location of the plant, and is sampled from a gaussian probability density function $N(T_{norm}(t), 0.1)$ with $T_{norm}(t) \in [-1{,}1]$ equal to the normalized monthly average temperature in the plant city [56] and $N_f(t) \in \{1,2,3,4\}$ is the number of components of the machine in abnormal conditions at time t.

The degradation of the four components of a machine is simulated using Eq. (10), until the machine reaches its mission time T_d or all four components start operating in abnormal conditions, which causes the machine failure. Figure 3 shows the simulated evolution of the degradation indicator of the four components of a machine and the operating condition $E(t)$ that they experience. Notice that this latter remarkably worsens when components 1 and 2 start operating in abnormal conditions.

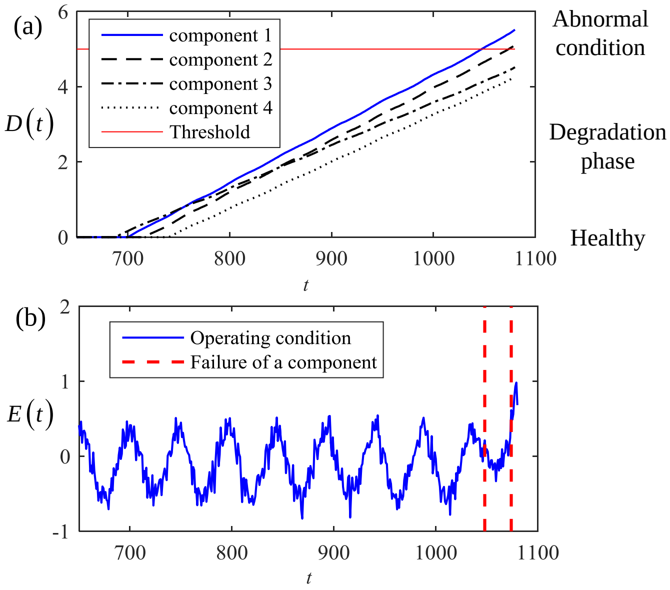

Figure 3. Evolution of: (a) the degradation indicators of the 4 machine components and (b) the machine operating condition $E(t)$

We further assume that the degradation states of the components cannot be directly measured, but ten sensors are installed on each component to measure quantities influenced by both the degradation $D(t)$ and the operating condition $E(t)$. The measurements are taken at a fixed frequency $f = 1$ atu^{-1}. The evolution of the 10 signals is simulated using the following equations:

$$\begin{aligned}
s_1(t) &= \sin(1/2\, D(t) + 2E(t)) \\
s_2(t) &= \sin(D(t) + \tan(E(t))) \\
s_3(t) &= \tan(1/2\, E(t) - 2/3\, w(t)) \\
s_4(t) &= \cos^3(w(t))\sin(E(t)) \\
s_5(t) &= \sin(3/10\, D(t) + 3w(t)) - E(t) \\
s_6(t) &= \sin(D(t) + w(t) - \tan^2(E(t) + 1)) \\
s_7(t) &= \sin(1/2\, D(t)E(t)) + 1/2\, w(t) \\
s_8(t) &= \cos^2(3/10\, D(t)E(t) - \tan(w(t))) \\
s_9(t) &= \cos(D(t) - \tan(E(t)) + 2\tan(w(t))) \\
s_{10}(t) &= \tan(1/4\, w(t)) - 1/4\, \cos(1/3\, D(t)\, \sin(E(t)))
\end{aligned} \tag{12}$$

Figure 4 shows an example of the time evolution of the 10 signals during a run-to-failure trajectory.

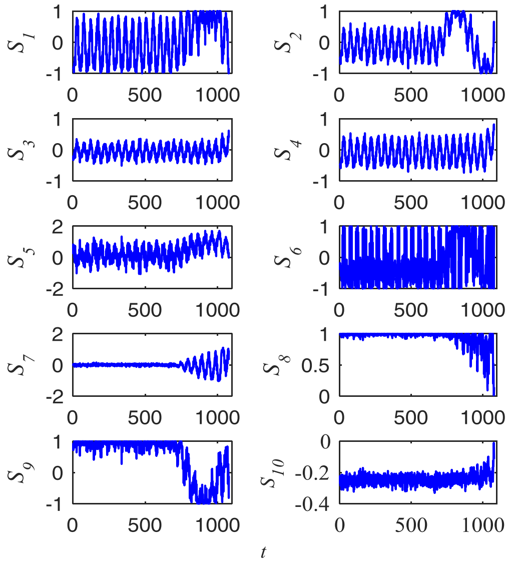

Figure 4. Signal measurements during a component run-to-failure trajectory.

The lives of 5000 machines have been simulated up to the first time between that of machine failure and the mission time. The obtained synthetic dataset is randomly divided into a training set and a test set, for the purposes of developing the SAE-DNN using the proposed method and of evaluating its performance, respectively. The training dataset $\{\mathbf{X}^{TRAIN}, \mathbf{Y}^{TRAIN}\}$ includes the measurements $\mathbf{X}^{TRAIN}$ and the corresponding 0/1 labels $\mathbf{Y}^{TRAIN}$ of the components normal/abnormal conditions collected from the simulation of 2500 machines; the test set is composed of the remaining data.

5.2. Application of the proposed method

A single run-to-failure trajectory is used to develop a stacked SAE, formed by an encoder with four layers of $K_1 = 100$, $K_2 = 50$, $K_3 = 20$, $K_4 = 3$ neurons and a symmetric decoder. Hyperparameters values are set following the procedure proposed in Section 4. The average computation time necessary to develop a stacked SAE using an Intel CPU E5-2640 v4 2.4 GHz 126.1 GB memory, Matlab R2017a 64 bit in Linux 2.6.32 is of about 54 seconds. Table 1 reports the identified best performing hyperparameters. The value of the threshold Th_{MK} is set using the method in Appendix A with a proportion $a = 0.7$. As described in Section 4, hyperparameters p, β and λ are at first set to $\bar{p} = 0.1$, $\bar{\beta} = 1$ and $\bar{\lambda} = 1e - 3$; then, p is tuned by randomly sampling values from an uniform distribution in the range $[0, 0.5]$ with fixed $\beta = \bar{\beta}$ and $\lambda = \bar{\lambda}$. Since a value $\rho_{SAE}(p, \bar{\beta}, \bar{\lambda}) \geq Th_{MK}$ has not been obtained by performing $N_{max} = 20$ samplings of p, p has been set equal to the value $p_{best} = 4e - 2$ at which $\rho_{SAE}(p, \bar{\beta}, \bar{\lambda})$ has its maximum. Then, λ is tuned by sampling $\log_{10}\lambda$ from the discrete uniform distribution $\{-20,-19,\dots,0\}$. Eventually, a satisfactory stacked SAE with $\rho_{SAE}\left(p_{best},\bar{\beta},\lambda\right) \geq Th_{MK}$ is found when $\lambda = \lambda_{best} = 1e-11$. Figure 5 shows the three features $\boldsymbol{z}^{(4)}(1)$, $\boldsymbol{z}^{(4)}(2)$ and $\boldsymbol{z}^{(4)}(3)$ extracted from the stacked SAE trained with the identified hyperparameters. Notice the monotonic trend of the feature $\boldsymbol{z}^{(4)}(1)$, which is characterized by $\left|\rho_{MK}\left(\boldsymbol{z}^{(4)}(1)\right)\right| = 25.57 > Th_{MK} = 25.08$.

Table 1. Stacked SAE hyperparameter values

Sparsity proportion	**Sparsity coefficient**	L_2 **regularization coefficient**
$\boldsymbol{p}$	$\boldsymbol{\beta}$	$\boldsymbol{\lambda}$
4e-2	1	1e-11

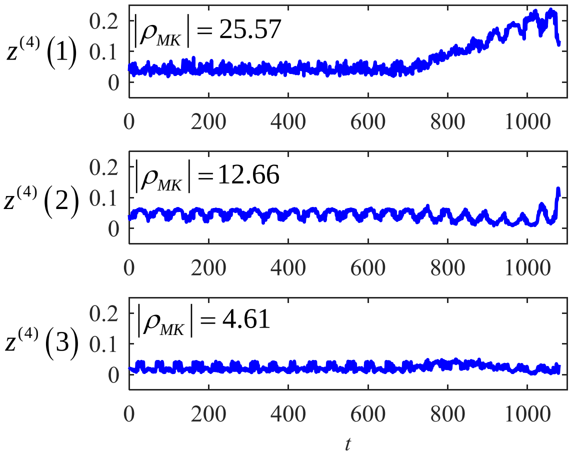

Figure 5. Features extracted from a single run-to-failure trajectory by the stacked SAE

Then, a SAE-DNN is built by adding to the stacked SAE a softmax layer with two output neurons and trained using the entire training dataset $\{\mathbf{X}^{TRAIN},\mathbf{Y}^{TRAIN}\}$.

The fault detection performance of the developed SAE-DNN is evaluated considering the 'false alarm' and 'missed alarm' rates on the test set:

$$\text{False alarm rate} = \frac{N^{M}_{normal}}{N_{normal}},\ \text{Missed alarm rate} = \frac{N^{M}_{abnormal}}{N_{abnormal}} \tag{13}$$

with N_{normal} indicating the total number of test patterns in normal condition, N^{M}_{normal} the number of normal condition patterns misclassified as abnormal condition, $N_{abnormal}$ the total number of test patterns in abnormal condition, $N^{M}_{abnormal}$ the number of abnormal condition patterns misclassified as normal condition.

Figure 6 shows the Receiver Operating Characteristic (ROC) curve [57] obtained considering the

SAE-DNN with values of Th_d (see Section 4) equal to $0.1, 0.2, \dots, 0.9$. The SAE-DNN fault detection performances are satisfactory in all cases, being the curve close to the bottom left corner, which indicates few false and missed alarms.

To investigate the effect of developing the stacked SAE using only one run-to-failure trajectory instead of all the available training data, another SAE-DNN is developed using the whole training dataset in both pre-training and fine-tuning phases. In this case, the computation time necessary for the pre-training phase of a single stacked SAE increases to 5350 minutes, but in the end the ROC curve is almost coincident to that of the SAE-DNN pre-trained using only one run-to-failure trajectory, which requires a computation time of only 54 seconds. This is due to the fact that the proposed method for the development of the stacked SAE, based on the use of a single run-to-failure trajectory, already allows extracting a high-level feature well-representing the degradation process, which is the key factor for the successful fine-tuning of the DNN. Notice that the use of all the run-to-failure trajectories would not allow, from the computational point of view, the application of the proposed procedure for hyperparameters tuning, which requires the development of multiple stacked SAEs.

The correlation between the monotonicity of the feature extracted by the stacked SAE and the fault detection performance of the corresponding, fine-tuned SAE-DNN has been investigated by training two other SAE-DNNs with the same architecture, but randomly selected hyperparameters. Figure 7 shows the missed and false alarm rates of the obtained SAE-DNNs when $Th_d = 0.5$, as a function of the monotonicity of the extracted features. The results confirm that larger the monotonicity of the extracted feature, the more satisfactory the fault detection performance. Therefore, setting the hyperparameters using the proposed procedure based on the maximization of ρ_{SAE} allows improving the fault detection performances of the SAE-DNN.

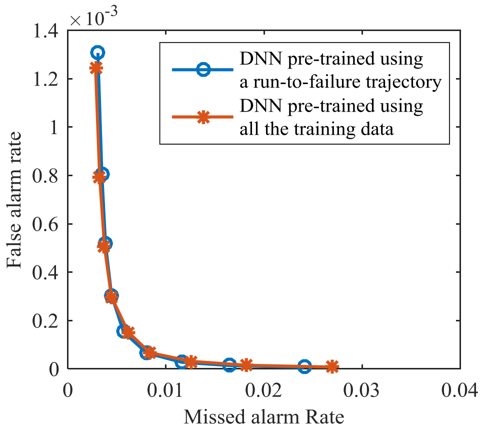

Figure 6. ROC curve of the SAE-DNNs trained using the proposed method (-o-) and all the training data (-*-)

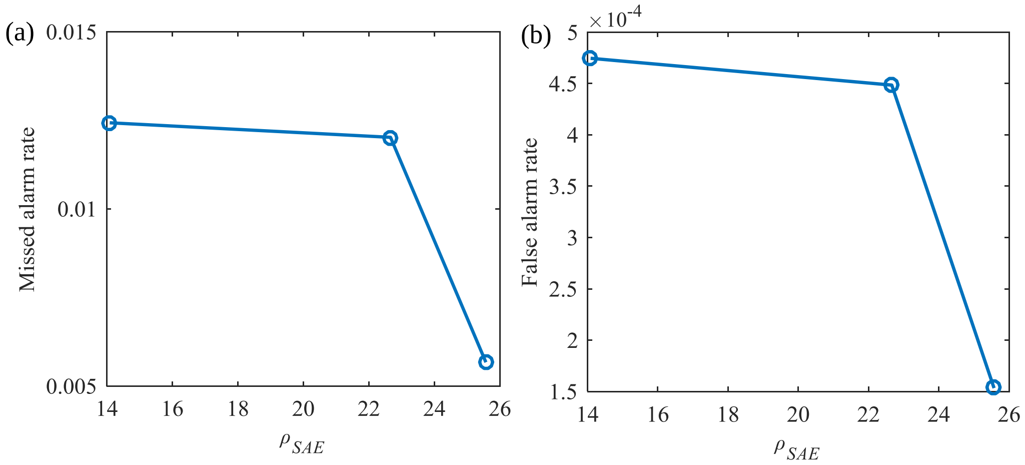

Figure 7. Missed and false alarm rates as a function of the monotonicity of the most monotonic extracted feature of the stacked SAE

The proposed method has been compared with fault detection methods based on ANN and ELM models. ANNs have been used for fault detection in many applications [13, 58, 59], whereas ELM is an emerging neural network which is used for its reduced training efforts and regression/classification accuracy comparable to those of ANNs [60]. An ANN model with layers of (10, 10, 1) neurons and an ELM model with layers of (10, 1000, 1) neurons and sigmoidal activation functions have been built using all the training dataset. The size of the ANN and ELM hidden layers has been optimized by trial-and-error. Figure 8 shows the obtained ROC curves together to that obtained with the SAE-DNN: the proposed SAE-DNN model is the best performing one, since it provides the ROC closest to the lower left corner of the ROC plane.

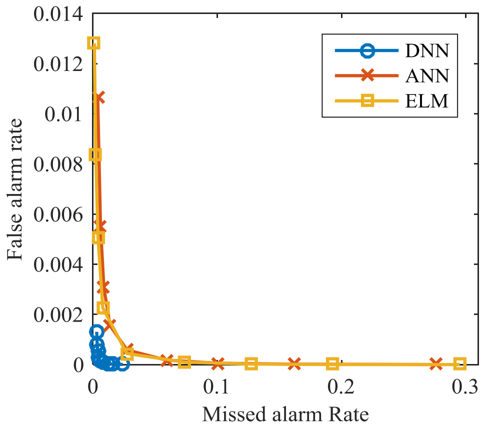

Figure 8. Comparison of fault detection results among the proposed SAE-DNN, ANN and ELM

6. Real case study

Bearings are the most critical components of rotating machinery [18]. Although several diagnostic systems have been proposed for bearing condition monitoring in evolving environments, they mainly consider variations of working loads and rotating speeds [14, 61-64], whereas few of them consider the effects of the degradation of one bearing on the other bearings of the same industrial equipment. In practical applications several bearings are typically installed on the same shaft and the degradation of one bearing causes large vibrations on the machinery, which considerably modify the operating conditions of the other bearings. Ignoring these effects can significantly reduce the performance of the fault detection system. In the following Sections, the proposed fault detection method is verified considering the monitoring of multiple bearings of the same rotating machinery, and the effects of the evolving environment that affects their operating conditions.

6.1. Data description

We consider the IMS bearing dataset [65], provided by the NSF I/UCR Center for Intelligent Maintenance Systems (IMS) at University of Cincinnati [43], which contains vibration data collected from a test rig hosting four test bearings installed on the same shaft (Figure 9). The shaft is driven by an AC motor and coupled by rub belts. During the tests, the rotation speed is kept constant at 2000 rpm, a radial load of 6000 lbs is applied onto the shaft and the bearings by a spring mechanism, and all bearings are force-lubricated. Accelerometers are placed closed to each bearing to measure vibrational signals. Snapshots containing 1-second *x*-axes acceleration measurements are collected every 10 minutes of operation, at a sampling frequency of 20,480 Hz.

We consider data collected from experiments E1 and E2, which are the most used in the literature [66-68]. Experiment E3 has not been considered given the lack of results for comparison. At the end of Experiment E1, bearings B3 and B4 are found failed, whereas at the end of experiment E2 bearing B1 is found failed (Table 2**)**.

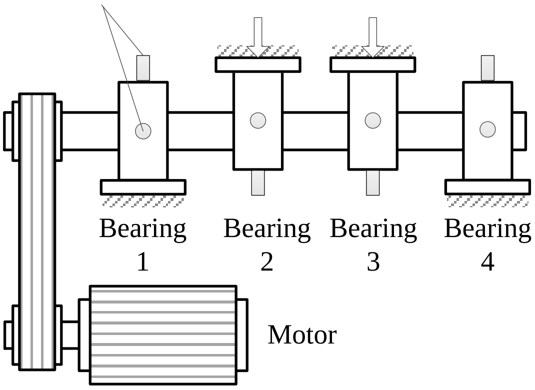

Figure 9. Bearing test rig [43]

Table 2. Description of the experiments

Experiment	Number of collected snapshots	Experiment duration (days)	Bearings found failed at the end of the experiment
E1	2156	35	B3 and B4
E2	984	8	B1

Since the exact time at which the bearing start operating in abnormal condition is not known, we consider the results of other studies [43, 66, 67] for labeling the dataset (Table 6 and Table 7**)**. In particular, the label 0 (normal condition) is assigned to all snapshots collected before the earliest detection, the label 0.5 (ambiguous classification) to all snapshots collected between the earliest and the latest detections, and the label 1 (abnormal condition) to all snapshots collected after the latest detection (Table 3).

Table 3. Dataset labels

Experiment	Bearing	Label associated to the snapshot		
		0	0.5	1
E1	B1	0~2156		
	B2	0~2156		
	B3	0~1617	1618~2027	2028~2156
	B4	0~1617	1618~1760	1761~2156
E2	B1	0~538	539~547	547~984
	B2	0~984		
	B3	0~984		
	B1	0~984		

6.2. Application of the proposed method

Features extracted from frequency spectra of the snapshots are used as inputs of the proposed fault detection method to reduce: *i*) the dimensionality of the 20480 acceleration values contained in a snapshot, which may lead to heavy computational burden and *ii*) the effects of phase shifts in the different snapshots, which may confuse the SAE-DNN. The frequency spectrum of a snapshot is obtained by applying the Morlet 6 continuous wavelet transform [69] and considering as features the time averages of the 1333 extracted frequency amplitudes. Morlet wavelet is used since its real part is a cosine signal that exponentially decays on both the left and right sides, with a function shape similar to an impulse, which makes this wavelet very attractive and widely applied in fault diagnostic applications [43]. Since the frequency spectrum of the acceleration signal is relatively stable during the 1-second duration of a snapshot, computing the time average of the time-frequency amplitudes allows reducing the input dimensionality without losing information.

The two different partitions of the available data into training and test sets reported in Table 4 are considered to verify the performance of the proposed method.

Table 4. The two considered partitions of the IMS bearing dataset

	Training dataset	**Testing dataset**
Case 1	E1	E2
Case 2	E2	E1

In both cases, a stacked SAE has been developed, composed of an encoder with four hidden layers of $K_1 = 1500,\ K_2 = 700,\ K_3 = 200,\ K_4 = 5$ neurons and a symmetric decoder.

6.2.1 Case 1

The stacked SAE has been trained using the run-to-failure data collected from bearing B3 in experiment E1. Stacked SAE hyperparameters values have been set using the procedure described in Section 4. The average computation time necessary to train a stacked SAE is 13.87 minutes.

Table 5. Parameters of the optimized stacked SAE

Sparsity proportion	**Sparsity coefficient**	$\boldsymbol{L_2}$ **regularization coefficient**
$\boldsymbol{p}$	$\boldsymbol{\beta}$	$\boldsymbol{\lambda}$
0.1	1	1e-5

The value of Th_{MK} is set to 25.00 using the method in Appendix A with a proportion $a = 0.8$. The values of the hyperparameters have been set by applying the procedure of Section 4, which has required the development of 36 stacked SAEs. The features $\boldsymbol{z}^{(4)}(1)$, $\boldsymbol{z}^{(4)}(2)$, …, $\boldsymbol{z}^{(4)}(5)$ extracted from the stacked SAE trained using the identified best performing hyperparameters (Table 5**)** are shown in Figure 10. Notice that feature $\boldsymbol{z}^{(4)}(1)$, which is characterized by the largest M-K coefficient, $\rho_{MK}\left|\boldsymbol{z}^{(4)}(1)\right| = 27.70$, shows almost constant values until snapshot 2000s, at which it starts monotonically increasing. Since this trend is typical of bearing degradation processes and is similar to that reported in [66] for the same bearing, it is possible to conclude that the stacked SAE has been able to extract a degradation indicator.

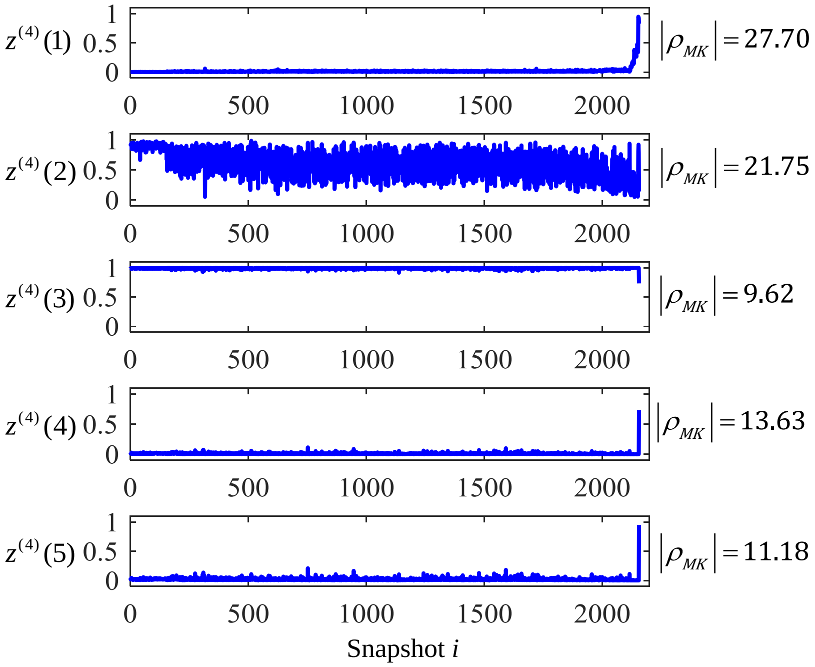

Figure 10. Features extracted by the developed stacked SAE

Then, the SAE-DNN is built using all the data collected in experiment E1 as training set, $\mathbf{X}^{TRAIN}$. The obtained SAE-DNN is tested using the data collected in experiment E2. Figure 11 shows the output of the softmax neuron, $\hat{y}_2{}^{soft}$, associated to abnormal condition. Notice that the SAE-DNN output firstly increases for bearing B1, for which a value of 1 is abruptly reached at snapshot 703, which is 281 snapshots before the end of the experiment, when the bearing was found failed. Also, the proposed method considers in normal conditions the remaining bearings B2, B3 and B4 for at least 200 snapshots after the detection of B1 abnormal condition. This result confirms the method robustness with respect to the evolving environment which, in this case, is caused by the effects of the vibration of bearing B1 on the other bearings degradation and on the sensor measurements. The increase of the SAE-DNN output for B2, B3 and B4 in the last part of the experiment can be attributed to: 1) a slow degradation process of the bearings which does not cause their failure before the end of the experiment; 2) the insufficiency of training data describing the effect of abnormal condition of B1 on the other bearings; 3) the increasing of the effect of B1 abnormal condition on the vibration of the other bearings. Applying a rule according to which the onset of abnormal condition is detected when the SAE-DNN output $\hat{y}_2{}^{soft}$ exceeds the threshold $Th_d = 0.5$ for at least three consecutive snapshots, the detected onsets of abnormal condition for all four bearings are reported in Table 6. Notice that the proposed fault detection method detects the abnormal condition onset of B1 with delays of 158 and 166 snapshots with respect to [66] and [67], respectively. This is due to the fact that the fault detection methods proposed in [66, 67] require the development of a specific model for each bearing, trained using historical data collected from that bearing at the beginning of its life when it is in normal conditions. Therefore, on the one hand, models in [66] and [67] allow promptly detecting the onset of abnormal conditions, on the other hand they cannot be generalized and applied to other new bearings and are less robust to evolving environments, since they have been developed using a limited amount of data representative of the bearing operational conditions at the beginning of its life. For these reasons, they are expected to provide large false alarm rates when used in evolving environments.

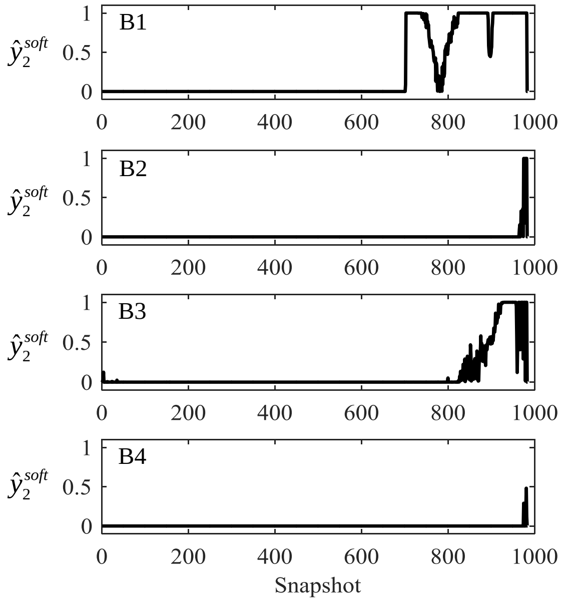

Figure 11. Case 1- Output of the SAE-DNN for experiment E2 data

Table 6. Snapshots at which the onsets of abnormal condition are detected for experiment E2. "-" indicates that the results have not been reported in literature, "/" indicates that abnormal condition has not been detected.

	Experiment E2			
	B1	B2	B3	B4
[66]	547	-	-	-
[67]	539	-	-	-
Proposed method	705	977	905	/
SAE-DNN pre-trained using all data	705	977	907	/
Proposed method-data collected in evolving environment are not used for SAE-DNN fine-tuning	705	967	874	982

6.2.2 Case 2

The stacked SAE has been trained using the run-to-failure data collected from bearing B1 in experiment E2. Following the procedure described in Section 4, the stacked SAE hyperparameters are set to the values reported in Table 5. The obtained SAE-DNN is tested using the data collected in experiment E1. Figure 12 shows the output of the softmax neuron $\hat{y}_2^{soft}$ associated to abnormal condition. Table 7 reports the time at which an abnormal condition is detected.

Notice that the first bearing for which $\hat{y}_2^{soft}$ exceeds the threshold value $Th_d = 0.5$ for three consecutive snapshots is B4, whose abnormal condition is detected at snapshot 1670, which is close to the detection snapshot reported in literature [43, 66, 67]. The bearing B3 abnormal condition is detected at snapshot 2122, with a delay of 95 snapshots with respect to [66], whereas B1 is considered in normal condition until the end of the experiment and B2 until two snapshots before the end. These results confirm the robustness of the method with respect to the evolving environment in the operating conditions of B1, B2 caused by B3 and B4 abnormal conditions.

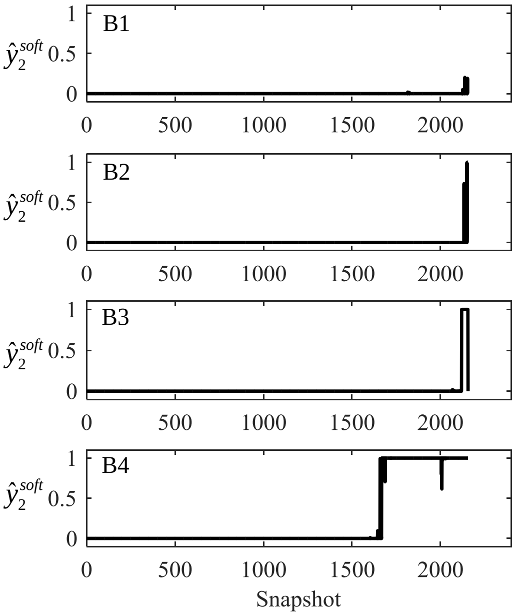

Figure 12. Case 2- fault detection results

Table 7. Snapshots at which the onsets of abnormal condition are detected for experiment E1. "-" indicates that the results have not been reported in literature, "/" indicates that abnormal condition has not been detected.

	Experiment E1			
	B1	B2	B3	B4
[43]	-	-	1617	1617
[66]	-	-	2027	1641
[67]	-	-	-	1760
Proposed method	/	2154	2122	1670
Proposed method- data collected in evolving environment are not used for SAE-DNN fine-tuning	2126	2128	2065	1670

6.3. Comparison with a SAE-DNN pre-trained using all the experiment data

A new SAE-DNN is pre-trained using all data collected in experiment E1 and compared to that developed in Section 6.2.1, which is pre-trained using only one run-to-failure trajectory. Table 6 reports the snapshots at which the abnormal conditions of the bearings in experiment E2 are detected. Notice that as for the synthetic case study of Section 5, the detection results are very similar in the two cases, although the computational time necessary to pre-train the SAE-DNN using all the data is much longer (43 minutes) than that of using only one run-to-failure trajectory (14 minutes).

6.4. Comparison with a SAE-DNN trained without considering data collected in the evolving environment

Two SAE-DNNs are developed excluding from the training set $\mathbf{X}^{TRAIN}$ the data collected after the modification of the environment caused by abnormal conditions in other bearings (Table 8).

Table 8. Labels assigned to the snapshots in the dataset excluding the effects of evolving environment

Experiments	**Bearing**	**Label to snapshots**		
		0	0.5	1
E1	B1	0~1617		
	B2	0~1617		
	B3	0~1617	1618~2027	2028~2156
	B4	0~1617	1618~1760	1761~2156
E2	B1	0~538	539~547	547~984
	B2	0~538		
	B3	0~538		
	B4	0~538		

In particular, a SAE-DNN is developed using all the experiment E1 run-to-failure data of B3 and B4 and only the snapshots 1 to 1617 of bearings B1 and B2, which have been collected before the onset of abnormal conditions on bearings B3 and B4 (Table 8). The obtained SAE-DNN is tested using all the data collected in experiment E2. Table 6 reports the snapshots at which abnormal conditions are detected. Notice that the abnormal condition detection is anticipated for bearings B2 and B3 with respect to that of the SAE-DNN trained using the proposed method. Furthermore, B4 is detected in abnormal condition, even if it is actually in normal condition until the end of the experiment. Similar results (Table 7) are obtained training the SAE-DNN using only the experiment E2 snapshots collected from B2, B3 and B4 before the onset of abnormal condition of B1 (Table 8). Therefore, it is possible to conclude that using all the available data for SAE-DNN fine-tuning allows building a more robust model with respect to evolving environments.

7. Conclusions

This paper contributes to addressing the problem of fault detection in evolving environments by offering a benchmark with synthetic data and a novel method of solution. The synthetic data benchmark is built with general characteristics of a detection problem in evolving environments and can serve for testing other future methods. The novel method for fault detection in evolving environments is based on the use of SAE-DNNs. Its main positive characteristics are the automatic extraction of a degradation indicator, without the necessity of designing handcrafted features, and its ability of performing fault detection in presence of evolving environments. Furthermore, a novel method for setting SAE-DNN hyperparameters during the SAE pre-training phase has been proposed.

The fault detection method has been verified also on a real case study. The results obtained in the case studies considered show that: *i*) a single run-to-failure trajectory allows identifying the best performing method hyperparameters, with an effective training of the stacked SAE and a considerable reduction of the computational effort, compared to using all training data; *ii*) the proposed method is able to minimize missed and false alarm rates in conditions of evolving environments and performed significantly better than the ANN-based and ELM-based fault detection methods considered for comparison. In particular, the results show that the proposed method can detect the degradation of a component, even if its behavior is strongly influenced by the presence of other degrading components in the system.

Appendix A

The value of the threshold Th_{MK} is set by estimating the monotonicity $|\rho_{MK}|$ of an artificial feature designed to represent the degradation trend in a typical run-to-failure trajectory of the same component. In industrial applications, the equipment typically works in normal condition for a long period of time, until there is the onset of the degradation process. Therefore, an ideal degradation feature extracted from a run-to-failure trajectory $\{\boldsymbol{x}_i^{RTF,q},\ i = 1, \dots, N_q\}$ should include a constant part representing the normal condition and a monotonic part representing the evolution from normal condition to abnormal condition (degradation):

$$\boldsymbol{z}_{ideal}(i) = \begin{cases} 0,\ i \leq \lfloor aN_q \rfloor \\ \frac{i-\lfloor aN_q \rfloor}{N_q-\lfloor aN_q \rfloor},\ \text{otherwise} \end{cases} \tag{A.1}$$

where $i = 1,2, \ldots, N_q$, $a \in [0, 1]$ is the proportion of the constant part of $\boldsymbol{z}_{ideal}$, which depends from the type of component. For simplicity, the monotonic part in Eq. (14) is modeled using a linear function, although, depending from the component and type of degradation process, it can follow other types of monotonic functions, such as quadratic or exponential. Notice, however, that the type of monotonic function has no effect on the computation of $|\rho_{MK}|$. Since the the feature extracted from the stacked SAE is noisy, the artificial feature is obtained by summing a Gaussian noise to $\boldsymbol{z}_{ideal}$:

$$\boldsymbol{z}_{\sigma}(i) = \boldsymbol{z}_{ideal}(i) + N(0, \sigma), \ \ i = 1,2, \ldots, N_q \qquad \text{(A.2)}$$

where σ is the noise standard deviation, which is set by estimating the noise standard deviation in the first 10% values of the most monotonic feature extracted from the stacked SAE. The value of the threshold Th_{MK}, which, therefore, depends from the specific stacked SAE, is obtained by simulating 100 possible $\boldsymbol{z}_{\sigma}$ evolutions with assigned a and σ, and computing the average value of the M-K coefficient. Figure shows the dependence of Th_{MK} from a and σ.

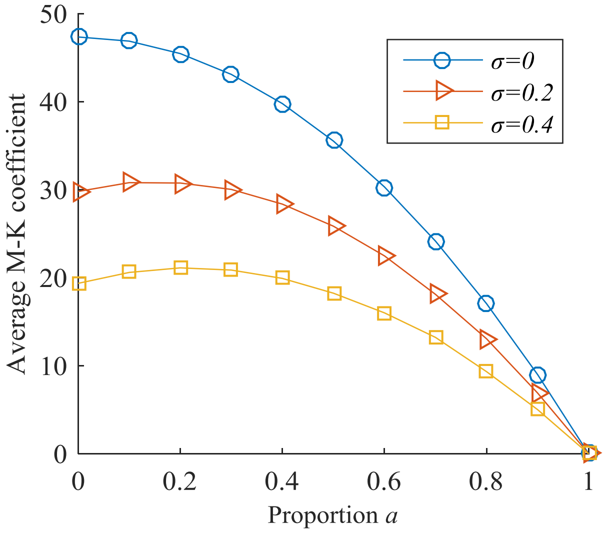

Figure A.1. Average values of the M-K coefficient of $\boldsymbol{z}_{\sigma}$ with different combinations of the parameters a and σ.

Acknowledgment

Zhe Yang gratefully acknowledges the financial support from the China Scholarship Council (No. 201506280015). We greatly appreciate the help from Dr. Marco Rigamonti for the literature review and sparse-overcomplete representation and the help from Mr. Pierluigi Colombo for the real case study. The work is developed within the research project "SMART MAINTENANCE OF

INDUSTRIAL PLANTS AND CIVIL STRUCTURES BY 4.0 MONITORING TECHNOLOGIES AND PROGNOSTIC APPROACHES - MAC4PRO ", sponsored by the call BRIC-2018 of the National Institute for Insurance against Accidents at Work – INAIL in Italy.

References

[1] J. Gertler, Fault Detection and Diagnosis, Springer, 2015.

[2] P. Baraldi, F. Di Maio, D. Genini, E. Zio, Comparison of data-driven reconstruction methods for fault detection, IEEE Transactions on Reliability, 64 (2015) 852-860.

[3] J. Yu, J. Yoo, J. Jang, J.H. Park, S. Kim, A novel hybrid of auto-associative kernel regression and dynamic independent component analysis for fault detection in nonlinear multimode processes, Journal of Process Control, 68 (2018) 129-144.

[4] F. Di Maio, P. Baraldi, E. Zio, R. Seraoui, Fault detection in nuclear power plants components by a combination of statistical methods, IEEE Transactions on Reliability, 62 (2013) 833-845.

[5] P. Baraldi, F. Di Maio, P. Turati, E. Zio, Robust signal reconstruction for condition monitoring of industrial components via a modified Auto Associative Kernel Regression method, Mechanical Systems and Signal Processing, 60 (2015) 29-44.

[6] E. Zio, F. Di Maio, A fuzzy similarity-based method for failure detection and recovery time estimation, International Journal of Performability Engineering, 6 (2010) 407-424.

[7] P. Baraldi, F. Di Maio, D. Genini, E. Zio, Reconstruction of missing data in multidimensional time series by fuzzy similarity, Applied Soft Computing, 26 (2015) 1-9.

[8] J. Tian, C. Morillo, M.H. Azarian, M. Pecht, Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with K-nearest neighbor distance analysis, IEEE Transactions on Industrial Electronics, 63 (2016) 1793-1803.

[9] P. Baraldi, F. Cannarile, F. Di Maio, E. Zio, Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable conditions, Engineering Applications of Artificial Intelligence, 56 (2016) 1-13.

[10] C. Titouna, M. Aliouat, M. Gueroui, FDS: fault detection scheme for wireless sensor networks, Wireless Personal Communications, 86 (2016) 549-562.

[11] R. Shrivastava, H. Mahalingam, N. Dutta, Application and Evaluation of Random Forest Classifier Technique for Fault Detection in Bioreactor Operation, Chemical Engineering Communications, 204 (2017) 591-598.

[12] J. Zheng, H. Pan, J. Cheng, Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines, Mechanical Systems and Signal Processing, 85 (2017) 746-759.

[13] P. Bangalore, L.B. Tjernberg, An artificial neural network approach for early fault detection of gearbox bearings, IEEE Transactions on Smart Grid, 6 (2015) 980-987.

[14] Y. Tian, J. Ma, C. Lu, Z. Wang, Rolling bearing fault diagnosis under variable conditions using LMD-SVD and extreme learning machine, Mechanism and Machine Theory, 90 (2015) 175-186.

[15] Z. Yang, P. Baraldi, E. Zio, A comparison between extreme learning machine and artificial neural network for remaining useful life prediction, 2016 Prognostics and System Health Management Conference (PHM-Chengdu), IEEE, 2016, pp. 1-7.

[16] P. Baraldi, R. Canesi, E. Zio, R. Seraoui, R. Chevalier, Genetic algorithm-based wrapper approach for grouping condition monitoring signals of nuclear power plant components, Integrated computer-Aided engineering, 18 (2011) 221-234.

[17] P. Baraldi, N. Pedroni, E. Zio, Application of a niched Pareto genetic algorithm for selecting features for nuclear transients classification, International Journal of Intelligent Systems, 24 (2009) 118-151.

[18] R. Liu, B. Yang, E. Zio, X. Chen, Artificial intelligence for fault diagnosis of rotating machinery: A review, Mechanical Systems and Signal Processing, 108 (2018) 33-47.

[19] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: A review, Data classification: Algorithms and applications, (2014) 37.

[20] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, 521 (2015) 436-444.

[21] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, I. Kompatsiaris, Deep learning advances in computer vision with 3d data: A survey, ACM Computing Surveys (CSUR), 50 (2017) 20.

[22] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems, 2012, pp. 1097-1105.

[23] L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related applications: An overview, Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 8599-8603.

[24] Y. Zhang, W. Chan, N. Jaitly, Very deep convolutional networks for end-to-end speech recognition, Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE, 2017, pp. 4845-4849.

[25] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal processing magazine, 29 (2012) 82-97.

[26] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, arXiv preprint arXiv:1708.02709, (2017).

[27] L. Deng, Y. Liu, Deep Learning in Natural Language Processing, Springer, 2018.

[28] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, science, 313 (2006) 504-507.

[29] A. Ng, Sparse autoencoder, CS294A Lecture notes, 72 (2011) 1-19.

[30] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 1096-1103.

[31] Z. Huijie, R. Ting, W. Xinqing, Z. You, F. Husheng, Fault diagnosis of hydraulic pump based on stacked autoencoders, Electronic Measurement & Instruments (ICEMI), 2015 12th IEEE International Conference on, IEEE, 2015, pp. 58-62.

[32] Y. Qu, M. He, J. Deutsch, D. He, Detection of Pitting in Gears Using a Deep Sparse Autoencoder, Applied Sciences, 7 (2017) 515.

[33] F. Jia, Y. Lei, J. Lin, X. Zhou, 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, 72 (2016) 303-315.

[34] Z. Chen, W. Li, Multisensor Feature Fusion for Bearing Fault Diagnosis Using Sparse Autoencoder and Deep Belief Network, IEEE Transactions on Instrumentation and Measurement, 66 (2017) 1693-1702.

[35] P. Chopra, S.K. Yadav, Fault detection and classification by unsupervised feature extraction and dimensionality reduction, Complex & Intelligent Systems, 1 (2015) 25-33.

[36] C. Lu, Z.-Y. Wang, W.-L. Qin, J. Ma, Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification, Signal Processing, 130 (2017) 377-388.

[37] I. Žliobaitė, Learning under concept drift: an overview, arXiv preprint arXiv:1010.4784, (2010).

[38] Z. Yang, S. Al-Dahidi, P. Baraldi, E. Zio, L. Montelatici, A Novel Concept Drift Detection Method for Incremental Learning in Nonstationary Environments, IEEE transactions on neural networks and learning systems, (2019).

[39] R. Izadi-Zamanabadi, M. Blanke, A ship propulsion system as a benchmark for fault-tolerant control, Control Engineering Practice, 7 (1999) 227-239.

[40] M. Syfert, R. Patton, M. Bartyś, J. Quevedo, Development and application of methods for actuator diagnosis in industrial control systems (DAMADICS): A benchmark study, IFAC Proceedings Volumes, 36 (2003) 843-854.

[41] M. Blanke, S.A. Bøgh, R.B. Jørgensen, R. Patton, Fault detection for a diesel engine actuator-a benchmark for FDI, Control Engineering Practice, 3 (1995) 1731-1740.

[42] H. Toubakh, M. Sayed-Mouchaweh, Hybrid dynamic data-driven approach for drift-like fault detection in wind

turbines, Evolving Systems, 6 (2015) 115-129.

[43] H. Qiu, J. Lee, J. Lin, G. Yu, Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics, Journal of sound and vibration, 289 (2006) 1066-1090.

[44] https://www.phmsociety.org/references/datasets.

[45] https://www.phmsociety.org/competition/phm/11.

[46] https://www.phmsociety.org/events/conference/phm/15/data-challenge.

[47] X. Jia, B. Huang, J. Feng, H. Cai, J. Lee, Review of PHM Data Competitions from 2008 to 2017: Methodologies and Analytics.

[48] https://www.phmsociety.org/events/conference/phm/17/data-challenge.

[49] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: A strategy employed by V1?, Vision research, 37 (1997) 3311-3325.

[50] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, nature, 323 (1986) 533.

[51] E. Doi, D.C. Balcan, M.S. Lewicki, A theoretical analysis of robust coding over noisy overcomplete channels, Advances in neural information processing systems, 2006, pp. 307-314.

[52] C. Poultney, S. Chopra, Y.L. Cun, Efficient learning of sparse representations with an energy-based model, Advances in neural information processing systems, 2007, pp. 1137-1144.

[53] Y.-l. Boureau, Y.L. Cun, Sparse feature learning for deep belief networks, Advances in neural information processing systems, 2008, pp. 1185-1192.

[54] T. Pohlert, Non-parametric trend tests and change-point detection, CC BY-ND, 4 (2017).

[55] C.M. Bishop, Pattern recognition and machine learning, springer, 2006.

[56] http://www.weatherbase.com/.

[57] J.V. Carter, J. Pan, S.N. Rai, S. Galandiuk, ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves, Surgery, 159 (2016) 1638-1645.

[58] Y. Maki, K.A. Loparo, A neural-network approach to fault detection and diagnosis in industrial processes, IEEE Transactions on Control Systems Technology, 5 (1997) 529-541.

[59] S.S. Tayarani-Bathaie, Z.S. Vanini, K. Khorasani, Dynamic neural network-based fault diagnosis of gas turbine engines, Neurocomputing, 125 (2014) 153-165.

[60] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: A review, Neural Networks, 61 (2015) 32-48.

[61] Y. Hu, P. Baraldi, F. Di Maio, E. Zio, A Systematic Semi-Supervised Self-adaptable Fault Diagnostics approach in an evolving environment, Mechanical Systems and Signal Processing, 88 (2017) 413-427.

[62] C. Cempel, M. Tabaszewski, Multidimensional condition monitoring of machines in non-stationary operation, Mechanical Systems and Signal Processing, 21 (2007) 1233-1241.

[63] M.S. Kan, A.C. Tan, J. Mathew, A review on prognostic techniques for non-stationary and non-linear rotating systems, Mechanical Systems and Signal Processing, 62 (2015) 1-20.

[64] R. Klein, E. Rudyk, E. Masad, Methods for diagnostics of bearings in non-stationary environments, International journal of Condition monitoring, 2 (2012) 2-7.

[65] A. Saxena, K. Goebel, C. Larrosa, F. Chang, CFRP Composites dataset, NASA Ames Prognostics Data Repository, 2015.

[66] R.M. Hasani, G. Wang, R. Grosu, An Automated Auto-encoder Correlation-based Health-Monitoring and Prognostic Method for Machine Bearings, arXiv preprint arXiv:1703.06272, (2017).

[67] J. Yu, Health condition monitoring of machines based on hidden Markov model and contribution analysis, IEEE Transactions on Instrumentation and Measurement, 61 (2012) 2200-2211.

[68] S. Kim, S. Park, J.-W. Kim, J. Han, D. An, N.H. Kim, J.-H. Choi, A New Prognostics Approach for Bearing based on Entropy Decrease and Comparison with existing Methods.

[69] C. Torrence, G.P. Compo, A practical guide to wavelet analysis, Bulletin of the American Meteorological society, 79 (1998) 61-78.