Degradation stage classification via interpretable feature learning

Antonio L. Alfeo [*], Mario G.C.A. Cimino, Gigliola Vaglini

Department of Information Engineering, University of Pisa, Largo L. Lazzarino 1, 56127, Pisa, Italy

ARTICLE INFO

Keywords:
Deep learning
Feature learning
Interpretable machine learning
Explainable artificial intelligence
Autoencoder
Predictive maintenance

ABSTRACT

Predictive maintenance (PdM) advocates for the usage of machine learning technologies to monitor asset's health conditions and plan maintenance activities accordingly. However, according to the specific degradation process, some health-related measures (e.g. temperature) may be not informative enough to reliably assess the health stage. Moreover, each measure needs to be properly treated to extract the information linked to the health stage. Those issues are usually addressed by performing a manual feature engineering, which results in high management cost and poor generalization capability of those approaches. In this work, we address this issue by coupling a health stage classifier with a feature learning mechanism. With feature learning, minimally processed data are automatically transformed into informative features. Many effective feature learning approaches are based on deep learning. With those, the features are obtained as a non-linear combination of the inputs, thus it is difficult to understand the input's contribution to the classification outcome and so the reasoning behind the model. Still, these insights are increasingly required to interpret the results and assess the reliability of the model. In this regard, we propose a feature learning approach able to (i) effectively extract high-quality features by processing different input signals, and (ii) provide useful insights about the most informative domain transformations (e.g. Fourier transform or probability density function) of the input signals (e.g. vibration or temperature). The effectiveness of the proposed approach is tested with publicly available real-world datasets about bearings' progressive deterioration and compared with the traditional feature engineering approach.

1. Introduction and motivation

Industry 4.0 advocates for the usage of machine learning and IoT technologies to automatically extract knowledge from industrial processes [1], drive technological innovation [2], and avoid production inefficiencies [3]. From the perspective of maintenance operations, the adoption of these technologies is enabling the transition from Reactive (RM) and Preventive Maintenance (PM) to Predictive Maintenance (PdM) [4,5]. With RM maintenance operations are executed if a failure occurs, thus it may result in production delay and high repair costs. PM aims at avoiding failures by carrying out maintenance operations according to a regular schedule and may result in unnecessary maintenance and high prevention costs [6]. PdM aims at providing a good trade-off between RM and PM, by planning the maintenance operations according to the estimated asset's health status and allowing the maintenance frequency to be as lower as possible. The health state of an asset can be obtained by processing its sensor data with artificial intelligence techniques, and even employed to predict the asset's remaining useful life (RUL). However, the reliability of RUL predictions may be affected by non-predictable and time-varying operational conditions, e.g. how intensively an asset is used while in "unhealthy" stage [7]. Thus, many real-world applications leverage health state estimation rather than RUL prediction. As an example, the authors in [8] derive a health state indicator by combining convolutional and recurrent neural networks, allowing the encoding of time-series information while generating the features to estimate the health state. If the asset's degradation behaves in a very consistent way, it can be modeled using a simple two-stage process, i.e. regular and unhealthy stage [9]. Otherwise, the unhealthy stage can be further divided to have a more accurate representation of the different behaviors characterizing the degradation process. Most of the research works [7] divide the degradation process into three [10], four [11], or five stages [12]. The number of stages used to characterize the degradation process is a design choice resulting from a trade-off between the interpretability of the prediction outcome and the complexity of the degradation process. A machine that has a more consistent degradation process can be effectively modeled with a few easy-to-interpret stages. Instead, an effective modelling of a complex degradation process requires a greater number of stages, but the

[*] Corresponding author.
E-mail address: luca.alfeo@ing.unipi.it (A.L. Alfeo).

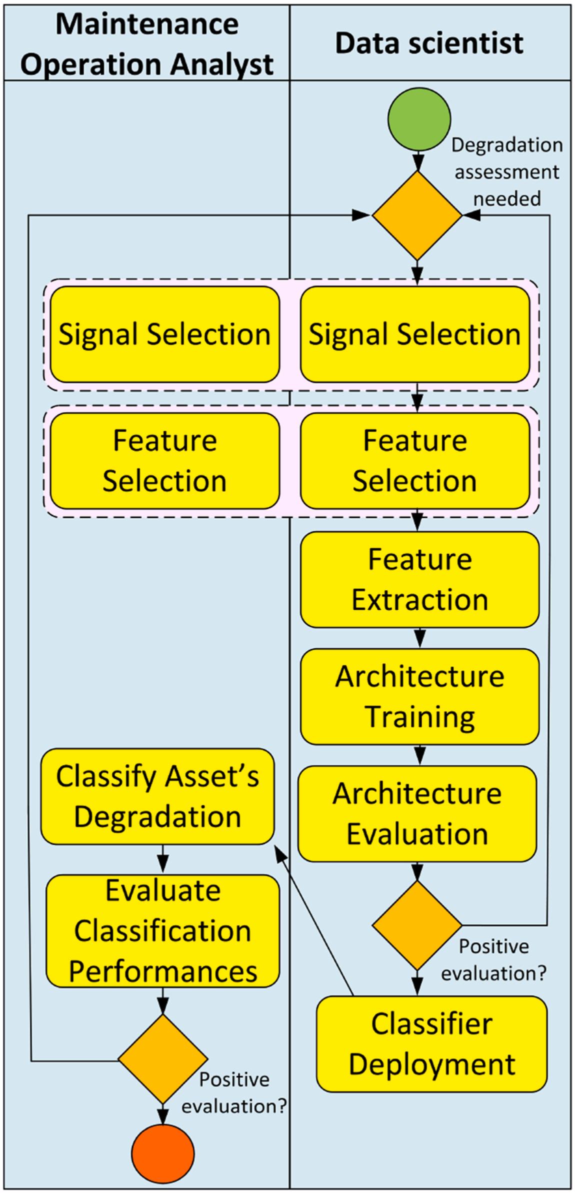


Fig. 1. Business Process Modeling Notation (BPMN) diagram of the setting-up of a predictive maintenance architecture.

meaning of each stage may result more difficult to be interpreted and less generalizable. The ability to generalize is one of the main issues with PdM approaches. Those tend to be designed and tailored for specific problems since the degradation processes may differ significantly across industries, plants, machines, and according to the sensors used to collect the measures to assess the asset's health condition [4]. To this aim, an example of possible measures may be temperature, vibration, power consumption, and noise [13]. However, some of these measures may be less informative than others according to the possible type of degradation (e.g. partial breakage or deterioration of a component) and the asset's operating condition (e.g. operating speed). Moreover, each measure needs to be properly treated to extract the information linked to degradation stage. These issues are usually addressed via a tight collaboration between data scientists and maintenance analyst to perform manual feature engineering, comprising (i) measure selection, according to the effect of a possible faults on the behavior of the signals; and (ii) feature extraction, i.e. transforming raw data into a more informative and compact representation, e.g. via statistical indicators. Due to (i) the multiplicity of the possible measures, (ii) the diversity of the degradation processes, and (iii) the specific expertise of the maintenance analyst involved, it is difficult to provide a high-quality feature engineering that is also generalizable among different PdM applications [14]. Moreover, manual feature engineering results in a high management costs also because this process may be repeated during the architecture training and evaluation, as well as every time the classification performance critically decrease, e.g. due to unseen operational conditions (Fig. 1). This results in the need to have a more automatic feature engineering process [15]. A possible solution is employing feature learning approaches [16], whose effectiveness have proved with many application domains such as speech recognition and object detection [17].

Feature learning has been studied since the advent of principal component analysis and has been recently revolutionized by the introduction of deep learning technology, resulting in better abstractions of the original input for subsequent classification, prediction, and detection tasks [18]. Contrary to feature engineering, feature learning is an integrated learning process: the algorithms learn to automatically transform minimally processed data into informative features able to simplify the classification task, reduce the management cost of a PdM architecture, and improve its performance. In this context, autoencoders (AE) are among the most popular deep learning architectures for feature learning [18,19]. An autoencoder is a deep neural network architecture made of two main components: an encoder and a decoder. The encoder is aimed at producing a compact representation of the inputs. The decoder is aimed at reconstructing the input data from a compact representation. The compact representation obtained by a properly trained autoencoder (i.e. the encoding) can be used as features for classification tasks. As feature learner, an autoencoder offers a few peculiar advantages [4], i.e. (i) by being trained in an unsupervised manner, they do not require prior knowledge about the data, and (ii) can fuse multi-sensory data while performing feature extraction [20,21]. On the other hand, the autoencoder cannot determine what information is relevant, it just learns how to extract meaningful non-linear combination of the inputs [4]. The capability to distinguish the most informative input is a property of manual feature engineering that is important to keep even in a feature learning approach. Indeed, it can be used [22] to (i) validate the model's decision-making against common knowledge or practitioner experience, (ii) provide managers with insights to identify potential causes of a short machine lifetime, and (iii) motivate the predictions obtained by processing those features. Indeed, with the wide adoption of machine learning technology, it has emerged how even models with high recognition scores may produce results that have no sense to a human observer [22]. Thus, machine learning models are increasingly required to provide not only a good recognition performance but also some insights about the model learned from the data, e.g. the inputs' contribution for the classification. Such insights are fundamental to interpret the model outcome and evaluate its reliability in a real-world scenario, especially with black-box approaches such as deep neural networks [23]. Specifically, in the field of predictive maintenance is emerging the need for interpretable approaches able to provide automatic feature learning from multiple and heterogeneous sources [7, 24,25], and an evaluation of the quality of the learned features, e.g. their relevance for the degradation stage classification [4]. We summarize the contribution of this work as it follows:

- We propose an approach to learn degradation-representative features from different sensory inputs, regardless of the nature of their time series (i.e. oscillatory or not) and regardless of which of them is more affected by the degradation process.
- The quality of the features learned from each input signal and domain transformation are evaluated by clustering them: the more the clusters match the degradation classes, the more the features are considered representative of the degradation process.

- Only the best learned features are used for the classification, thus their rank describes their contribution in the classification, providing global interpretability to the model.

To the best of our knowledge, this is the first example of a feature learning approach able to effectively process different input signals regardless of whether their behavior is oscillatory or not, thanks to a ranking and selection mechanism of the learned features based on their expected contribution in the classification. The proposed approach has been tested on 3 real-world cases study addressing the temperature and vibration of rolling bearings. The paper is structured as follows. In section 2, we present the literature review. Section 3 details our approach. The case study and the experimental setup are presented in sections 4. Finally, section 5 and 6 discuss the obtained results and the conclusions, respectively.

2. Related works

Feature learning approaches can employ both *linear* and *non-linear* methods [26], and more recently *deep neural networks* [19]. The mostly used *linear* methods are principal component analysis (PCA), factor analysis, and linear discriminant analysis. PCA [27] is a statistical technique aimed at finding the principal components of inputs hyperspace, i.e., the directions that maximize the variance between data points while being uncorrelated with the other components. The projections of a given data point over these directions can be used as features of that data point. Factor analysis [28] linearly combines a set of latent variables or unobserved factors to generate the features. Linear discriminant analysis [29] is a supervised statistical technique aimed at finding linear combinations of features to better distinguish different classes.

Some well-known *non-linear* approaches are manifold learning methods, kernel PCA, and restricted Boltzmann machines. Manifold Learning methods try to generalize linear frameworks (e.g. PCA) looking for non-linear lower-dimensional structures embedded in data. As an example, multidimensional scaling (MDS) aims at projecting samples in a low-dimensional space while preserving samples' pairwise distances. Kernel PCA [14] uses kernels to extend PCA with a non-linear combination mechanism and project data samples onto higher-dimensional spaces [30]. Restricted Boltzmann machines [31] is a generative stochastic artificial neural network able to learn inputs' probability distribution and use it to generate features from input data.

Deep neural network (DNN) architectures consist of hierarchies of abstractions of the input data whose relevant information is captured, combined, and passed to the next layer to be transformed in a proper result in the output layer. The features can be obtained by considering the information exchanged among the layers of specifically trained DNN, such as Generative Adversarial Networks [32], Deep Belief networks, and more often, AEs [33]. AEs employ a sort of "information bottleneck" to learn lower dimensionality representations of original inputs and faithfully reconstruct the input from that representation [24]. The generation of this representation (i.e. the encoding) can be constrained to provide it with specific properties, such as robustness to input noise (denoising autoencoder), enhanced organization of the latent space (variational autoencoder), or better compression capability (stacked autoencoder) [34]. By being descriptive enough to enable the input reconstruction, the AE's encoding can be considered as a feature learned from the input data. It is indeed used in this manner in many research works analyzing machinery's health condition. As an example, in [35] the authors leverage a feature learning and fusion mechanism obtained with denoising and a contractive AE with an approach for machinery fault diagnosis, obtaining 0.97 accuracy score. Authors in [36] propose an approach based on AE to detect faulty conditions in gearboxes and locomotive bearings, resulting in an accuracy equal to 0.94 and 0.89, respectively. A stacked denoising AE was deployed in [37], to adaptively extract features for health condition detection from vibration time series, resulting in 0.94 accuracy. An approach based on sparse AE was proposed in [38] to perform condition monitoring of an air compressor and achieving an accuracy up to 0.97. Similar performances are achieved in [39], in which an architecture based on sparse AE is employed for fault diagnosis of induction motor. Authors in [40] propose a deep learning framework for the degradation process monitoring leveraging a novel eigenvector based on time–frequency-wavelet joint features processed via a deep autoencoder. In [41], a stacked multi-level denoising autoencoder is employed to learn robust and discriminative features to detect fault in wind turbine gearbox, resulting in the maximum accuracy of 0.98. Authors in [42] propose an approach based on variational autoencoder to learn representative bearings' degradation features to be used for an effective health state estimation. In [43] the authors derive an operation-specific health indicator from industrial condition monitoring data via a generative deep learning model based on the conditional variational autoencoder. The most common approach with autoencoders operating on multiple input sources is trying to obtain a (fully or partially) shared encoding [44], allowing for the reconstruction of all the inputs data from it [34,45]. The main problem with this approach lies in the difficulty to extract some knowledge about the importance of each information source, given that their compact representation is made by their joint non-linear combination. As already mentioned in Section 1, the aspect of the explainability of ML approaches is crucial and well known in the literature. Explainable machine learning groups the machine learning approaches able to provide insights about the reasoning behind their outcome. An explanation of a model can be evaluated according to its interpretability and completeness [46]. The goal of interpretability is to provide human-understandable insights about the mechanism used by the model to produce a result, e.g. the contribution of each input in the prediction. The goal of completeness is to accurately describe each operation performed by the model to transform input data in predictions, e.g. the formulae expressing the data processing provided by a neural network. An explanation is complete when it allows to anticipate the behavior of the model in each situation [46].

Deep neural networks are black box model, thus are more complex to explain in an interpretable way [47]. Still, this can be achieved via approaches belonging to one of the following families [46], i.e. *representation-based, processing-based,* and *explanation-based.* Approaches considering the *representation* of information in the model aim at interpreting the prediction by examining the role of each neural network layers, neuron units, and latent space vectors' direction. To interpret a model according to how it *processes* data, two strategies are possible: (i) producing a saliency map [48], e.g. repeatedly testing the neural network with portions of the input occluded to create a map showing which parts of the data affected the network output; and (ii) treating the original model as a black-box and using a surrogate model, i.e. a model that behaves similarly to the original one but is easier to explain, i.e. via LIME [49] or SHAP [50]. LIME [49] (Local interpretable model-agnostic explanations) employs local surrogate models to provide insights about the contribution of each input in the model. The explanations provided by SHAP (SHapley Additive exPlanations) [50] works similarly, but they come with theoretical guarantees about their consistency and local accuracy. SHAP considers the contributions of all permutations of all the features of the model, whereas LIME perturbs data around an individual prediction, resulting in lower computational costs. On the other hand, LIME assumes linear behavior of the machine learning model locally, and may result in instability of the explanations, i.e. the explanations of two very close instances may differ significantly [51]. Both LIME and SHAP have been proposed as methods to provide local interpretability, i. e., explaining individual predictions rather than the model at the global level. However, LIME can provide also global interpretability via its *submodular pick* algorithm. It selects a set of representative data instances, i.e. whose non-redundant local explanations can be used to explain the model from a global perspective [49]. Still, it is unclear (i) to what extent the chosen instances are representative of the global

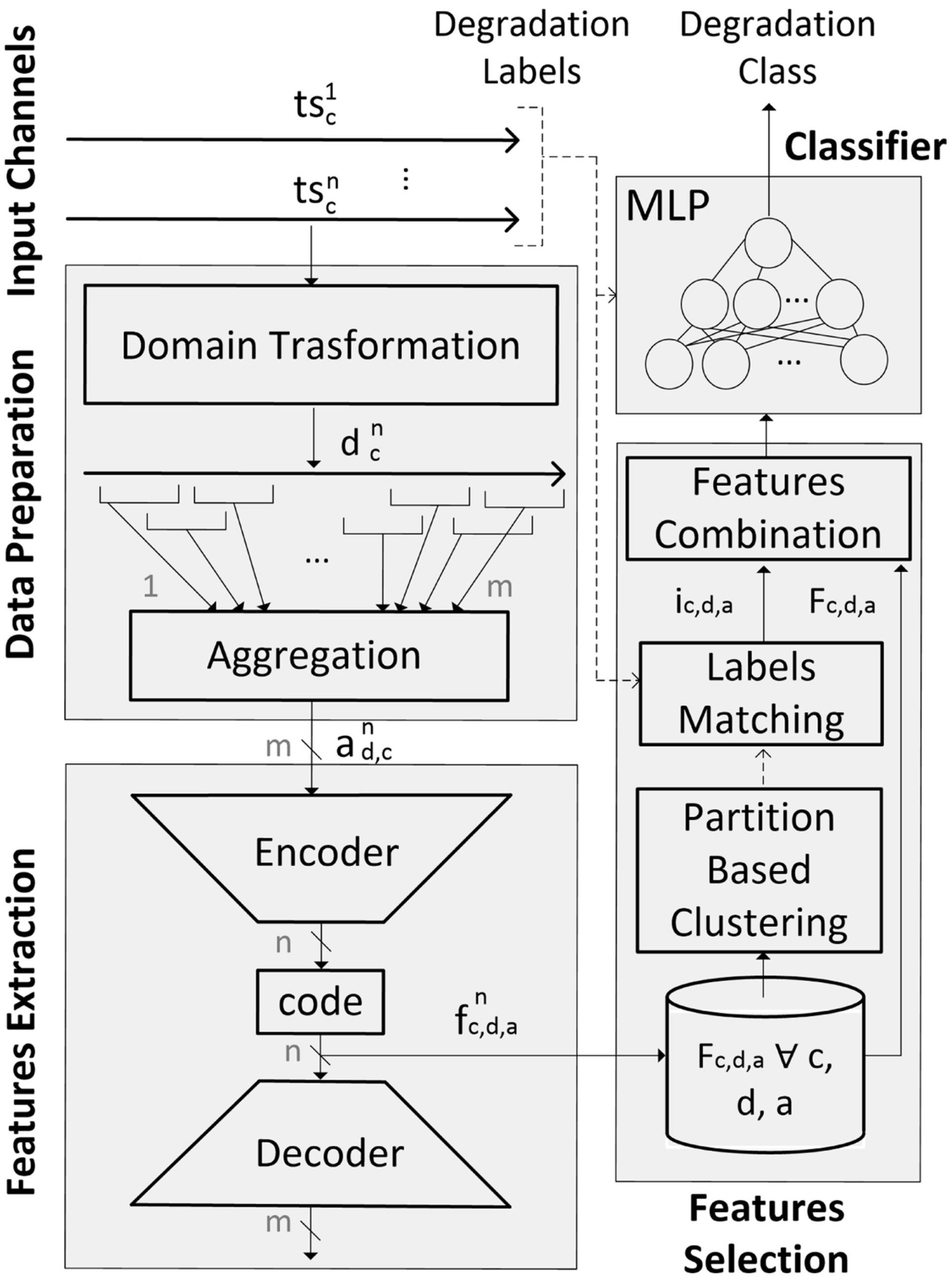


Fig. 2. The architecture of the proposed approach.

behavior of the model, (ii) how to identify the number of instances to have an effective global understanding of the model [52], as well as (iii) how to evaluate the trustability of a linear surrogate model when the object of the explanation represents complex non-linear relationships between input and output, as in the case of deep learning approaches [53]. To better interpret the behavior of these approaches, some *explanation* may be generated during the training of the deep neural network itself, via (i) attention-based networks, i.e. providing a weighting over inputs and internal features to steer the relevant information or the most engaged part of a network; (ii) neural networks providing data disentangled representations, i.e. organizing the data in the latent space to match the distribution of semantically meaningful factors of variation in the data, e.g. measure and trend of a time series [54]; and (iii) neural networks explicitly designed to generate their own explanation during their training. With *explanation-based* approaches, the interpretability of the model may result in higher model complexity since part of the training process is explicitly engineered to provide an additional outcome, i.e. some interpretability to the results [55]. Different *processing-based,* and *explanation-based* approaches have been used to interpret approaches based on AEs [19]. An example of a *processing-based* method is used in [56] where an AE-based architecture is used to perform feature extraction with multiple information sources, whereas the interpretation is addressed with an approach inspired to LIME approach. In contrast with *processing-based, explanation-based* approaches result to be more suitable for human evaluation [46] and therefore usable in an industrial process such as a predictive maintenance. As an example, in [57] authors obtain a disjointed multi-source representation in the latent space, allowing to consider the contributions of the single sources independently. Given its simplicity and effectiveness, a similar approach is also employed in this research work.

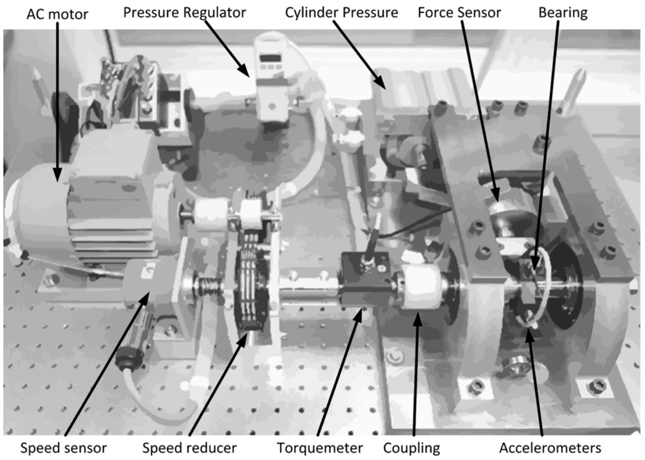


Fig. 3. Representation of the *Pronostia* platform.

3. Architecture design

In Fig. 2, we present the design of the proposed approach. It consists of four functional modules, addressing data preparation, feature extraction, feature selection, and health status classification. Neural networks are known as universal approximators. However, some generic well-known input transformations are ineffective or inefficient to be obtained through neural networks. For this reason, it is often preferred to pass already transformed data to the network, e.g. the Fourier transform [58] or the spectral kurtosis [59] of a given time series. As such, the first step of the data preparation is a domain transformation. Multiple domain transformations can be provided to work with time series consisting of different type of input signals. Each domain transformation turns a time series generated by one input channel ts_c in a numeric array d_c. Those arrays are split into m semi-overlapping segments and each segment is summarized by an aggregation operator, e.g. their average. Finally, the *data preparation* module processes n multivariate time series to generate a number of sets consisting of n m-length arrays $a_{d,c}$ one for each combination of transformation functions d and aggregation operators a, and input channel c. The set of the arrays $a_{d,c}$ obtained from a given channel c, with a domain transformation d and an aggregation operator a is rescaled between 0 and 1 with a min-max procedure and used as input for the *feature extraction* module. The feature extraction module consists of an autoencoder, trained to minimize the input reconstruction error, and used to generate representative features (i.e. the AE's encoding) from the input data. This feature extraction procedure is repeated for each possible combination of c, d and a. The resulting features are collected and passed to the *feature selection* module.

Our approach does not apply any aprioristic selection of the domain transformations based on the behavior (oscillatory or not) of the input signal; instead, it ranks the features learned from each input signal and domain transformation, to select only the best ones. Indeed, some input signals may be less affected by a given degradation process. At the same time, depending on the type of input signal, some domain transformations could not result in representative learned features and therefore increase the noise in the classification task. For instance, the Fourier transform of a vibration signal is expected to inform about the degradation of machinery, thus should result in representative learned features. On the other hand, the same transformation applied to the temperature signal may result in noisy non-informative features from the classification perspective. Specifically, the proposed approach provides both transformations for oscillatory and non-oscillatory signals. This choice is motivated by the fact that predictive maintenance is most often based on input signals that belong to one of these two categories, such as vibration, temperature, environmental noise, and energy consumption associated with the monitored equipment [7,24]. A selection mechanism is therefore essential because it is very likely that from each time series both representative and noisy features will be learned. The *feature selection* module aims at identifying the most informative combinations of channel c, domain transformation d, and aggregation operator a for a given classification task. The best combination can be identified according to the distance between the generated features, which should be shorter within the same class and greater among different classes. To this aim, the set of features obtained with each combination of c, d and a undergoes a partition-based clustering procedure, i.e. K-means [60]. The best quality combination is the one whose resulting cluster labels match the original class labels. Considering that the labels generated by the clustering may not have the same arrangement of classes' labels, this match is evaluated by using the adjusted rand index (ARI). The adjusted rand index can be roughly defined as an accuracy measure that considers possible labels' permutation [61]. ARI is bounded between 1 and 0. A greater ARI corresponds to a better match between the labels generated by clustering and the labels of health classes. Only the features with higher ARI are considered for the classification. The selection of the learned features is specific for each input signal and domain transformation, thus providing some insights into their contribution to the classification model. This result in model interpretability and comes with no additional costs, since (i) the ranking of the learned feature is part of the training process and does not require a posteriori computation to evaluate the contribution of the learned features in the classification, (ii) the selection of the learned features results in a simpler classification task; without it, a more complex end-to-end deep network would be needed to cope with such a noisy data and achieve good classification performance, and (iii) does not result in additional parameters to the model, by being based on a clustering procedure that does only require the number of clusters, which corresponds to the number of degradation stages. Finally, the features obtained with the best n combinations (even from different channels) are concatenated to generate the *features learned* from the data, i.e. the inputs for the degradation state *classifier*. The classifier employs a multilayer perceptron [62] able to process the features learned from the data and recognize the corresponding machinery health status, available as labels of the original time series.

Table 1
Number of time series per case study and health stage.

Case Study	# Regular	# Degraded	# Critical
A	1871	1665	181
B	748	319	74
C	753	371	73
D	5850	2834	3730

Table 2
CI of the F1-scores obtained by varying the number of best encodings used to generate the features. The best performances per case study and feature extractor are highlighted in bold.

Features Learner	# best enc.	Case Study A	Case Study B	Case Study C	Case Study D
AE	4	**0.998 ± 0.0007**	0.954 ± 0.0072	0.878 ± 0.0106	0.852 ± 0.005
	6	0.998 ± 0.0008	**0.968 ± 0.0066**	0.908 ± 0.0111	0.905 ± 0.005
	8	**0.998 ± 0.0007**	0.966 ± 0.0057	**0.947 ± 0.009**	**0.912 ± 0.003**
VAE	4	0.991 ± 0.0025	0.924 ± 0.012	0.836 ± 0.012	0.764 ± 0.005
	6	0.993 ± 0.0021	0.941 ± 0.0085	0.874 ± 0.0106	0.766 ± 0.005
	8	**0.994 ± 0.0022**	**0.950 ± 0.0068**	**0.913 ± 0.013**	**0.798 ± 0.010**

Table 3
Average F1-score degradation due to the lowering of the training epochs of the classifier, from 500 to 100.

Features Learner	Case Study A	Case Study B	Case Study C	Case Study D
AE	0%	−2%	−6.8%	−8.1%
VAE	0.3 %	−1.5%	−4.6%	−1.1%

4. Case study and experimental setup

The data used as case studies are collected via the experimental platform *Pronostia* and publicly available [63]. It enables validating approaches for bearing health assessment by leveraging some run-to-failure time series. Those time series are collected during the progressive degradation of bearings, resulting from the application of a radial force through an actuator (Fig. 3). In three case studies [63] the bearing's health is monitored with two types of signals: (i) temperature, sampled at 10 Hz, and (ii) vibration, sampled at 25.6 kHz with horizontal and vertical accelerometers. Since the time series of the vibration on the horizontal and vertical axis are strictly correlated, only the first one is considered. Those three case studies are named 1_1, 1_2, and 2_1 in [63]. For simplicity, in this work, we refer to them as case A, B, and C, respectively. We also employ another dataset [64], consisting of the vibration signals of a wind turbine high-speed shaft driven by a 20-tooth pinion gear, collected during 50 consecutive days. An inner race fault developed and caused the failure of the bearing across the 50 days. By concatenating these time-series we obtained one single run-to-failure time series, suitable for our analysis. We refer to this dataset as case study D.

First, those long run-to-failure time series are broken down into partially overlapping time windows, each of them from now on is called time series for simplicity. The duration of each of them is equal to 30 s and corresponds to a health stage of the bearing. In this study, we consider 3 health stages: (i) regular, in which the bearing operates normally and there is no evidence of degradation, (ii) degraded, i.e. characterized by more and more evidence of health degradation, and (iii) critical, in which the bearing is close to failing. To determine the degradation labels for each time series, we consider the instants in which the bearing may be considered out of the regular health stage or transitioning to the critical health stage. According to [65,66], a bearing can be considered at the beginning of its degradation process when the acceleration of the vibration signal is consistently equal or greater than 1 g. Thus, we smooth the vibration time-series via a moving average and consider the time instant in which the vibration is greater than 1 g as the first instant of the degraded health stage. To detect the transition to the critical health stage, we observe how quick Root Mean Square (RMS) of the vibration raw signal increases [67]. Specifically, the RMS curve is approximated via a polynomial regression, and the time instant in which the slope difference between two consecutive instants exceeds its 95th percentile is considered the beginning of the critical health stage. The resulting number of examples for each case study and health stage are reported in Table 1.

Table 4
CI of the F1-scores obtained by using the feature extractor and the handcrafted features presented in Section 4. The best performances are highlighted in bold.

Case study	Signal	Hand-crafted features	AE	VAE	[78]
A	Vibr.	0.830 ± 0.0249	**0.998 ± 0.0009**	0.995 ± 0.0014	0.968 ± 0.0019
	Temp.	0.941 ± 0.0143	0.955 ± 0.0038	**0.975 ± 0.0032**	–
	Both	0.968 ± 0.0118	**0.999 ± 0.0006**	0.994 ± 0.0022	–
B	Vibr.	0.755 ± 0.0545	**0.951 ± 0.0076**	0.828 ± 0.0135	0.783 ± 0.0120
	Temp.	0.817 ± 0.0348	0.877 ± 0.0098	**0.898 ± 0.0095**	–
	Both	0.907 ± 0.0253	**0.966 ± 0.0057**	0.950 ± 0.0068	–
C	Vibr.	0.757 ± 0.0505	**0.958 ± 0.0070**	0.924 ± 0.0100	0.674 ± 0.0080
	Temp.	0.879 ± 0.0437	0.918 ± 0.0066	**0.921 ± 0.0062**	–
	Both	0.882 ± 0.0599	**0.947 ± 0.0091**	0.913 ± 0.0130	–
D	Vibr.	0.899 ± 0.0147	**0.912 ± 0.0032**	0.798 ± 0.010	0.856 ± 0.0038

The input channels in our case studies consist of bearing's vibration and temperature signals. Vibration time series are usually analyzed in the frequency domain, whereas an analysis in the time domain can be sufficient with less fluctuating time series such as the temperature. Since the proposed approach is supposed to work with these two different types of input signals, the data preparation module is set to provide the following general-purpose transformations: probability density function, discrete Fourier transform, and spectral kurtosis. Considering that the Fourier transform has a real and an imaginary part, each time series ts_c is used to generate five numeric arrays d_c: the series in the time domain, its probability density function, its discrete Fourier transform (real and an imaginary part), and its spectral kurtosis. Each one of those arrays are split in 128 semi-overlapped segments. Finally, each segment is aggregated via their mean or standard deviation. The autoencoder used in the feature extractor consists of 4 dense neural network layers for the encoder (128 + 64 + 32 + 16 neurons) and the same for the decoder (16 + 32 + 64 + 128 neurons). As loss function, we use the mean square error (MSE), as neurons' activation function we use *relu* (rectified linear unit), and *adam* as optimization strategy due to its computational efficiency and little memory requirements [68]. Beside the "basic" autoencoder, the variational autoencoder could be useful in a feature learning problem, as it attempts to constrain the latent space to have a better spatial organization of the encodings, i.e. with a Gaussian distribution [69]. Therefore, we compare deep autoencoder and variational autoencoder in our experimentations. Both feature extractor models have the same number of neurons with the only exception of the last encoder layer, which is generative for the variational autoencoder [34]. *K-means* clustering algorithm is used for the feature selection module. The classifier consists of a multiplayer perceptron with 3 hidden layers, each one of 16 neurons with *relu* activation function. Each architecture module is built in Python, by using well known machine learning libraries (e.g. *sklearn* and *tensorflow)*. To assess the benefits of a feature learning approach with respect to a classic feature extraction approach, we employ a set of largely used features for industrial assets' degradation analysis [3,70]. Specifically, we extract:

(a) 90th, 75th, 50th, and 25th percentile of the time series
(b) maximum, median, mean absolute deviation, skewness of the time series
(c) the difference between the global (i.e. of the whole run-to failure time series) and local (i.e. of the current time window) mean absolute deviation
(d) the difference between the global and local median [3]
(e) number of continuous time-intervals with values greater than 90th, 75th, 50th, and 25th percentile of the time series [3]
(f) number of samples greater than 50 % and 25 % of the maximum of the time series [3]

Table 5
CI of the F1-scores obtained by processing the features extracted with K-means.

Case study	Handcrafted features	AE	VAE
A	0.909 ± 0.0038	**0.974 ± 0.0230**	0.901 ± 0.0249
B	**0.584 ± 0.0094**	0.512 ± 0.0303	0.511 ± 0.0249
C	0.622 ± 0.0256	**0.759 ± 0.0532**	0.646 ± 0.0406
D	0.389 ± 0.0070	**0.643 ± 0.0229**	0.559 ± 0.0348

Table 6
F1-scores CI obtained with a VAE-based approach, by adding white Gaussian noise to the vibration signals of the Pronostia dataset. The white Gaussian noise has been parametrized according to the signal-noise ratio (SNR).

SNR	CASE A	CASE B	CASE C	CASE D
0.7	0.957 ± 0.021	0.786 ± 0.039	0.811 ± 0.034	0.721 ± 0.097
0.8	0.984 ± 0.011	0.813 ± 0.063	0.864 ± 0.051	0.749 ± 0.095
0.9	0.988 ± 0.009	0.815 ± 0.041	0.841 ± 0.040	0.780 ± 0.068
1	0.981 ± 0.018	0.833 ± 0.059	0.870 ± 0.036	0.760 ± 0.043

Table 7
CIs of the execution time of a single training epoch (in seconds).

Training time [sec]	CASE A	CASE B	CASE C	CASE D
AE	17.9 ± 1.1	10.2 ± 0.6	10.1 ± 0.6	40.2 ± 3.4
VAE	19.7 ± 0.7	12.8 ± 0.6	13.4 ± 0.6	30.0 ± 1.6
MLP	2.0 ± 0.3	1.7 ± 0.1	1.9 ± 0.2	2.8 ± 0.3

(g) root mean square, crest factor, impulse factor, peak to peak, entropy, kurtosis of the time series [70]

Some features are extracted only with temperature data (e, and f), others only with vibration data (g). The results obtained by passing those features to the degradation classifier, should allow us to know if the proposed feature learning approach is able to compensate for the lack of a traditional feature engineering.

We evaluate the capability of the feature selection module of (i) filtering out noisy or non-informative learned features for classification, and (ii) generating a global explanation of the proposed method. Thus, we repeat the measurement of the classification performances without the feature selection module, by just connecting the features provided by all the trained autoencoders to the MLP classifier. To test whether a more powerful or better-trained classifier could compensate for the increase in classification noise, we test many different hyper-parameterizations of the classifier via a Bayesian optimization approach, a global optimization method for noisy black-box functions [71]. Specifically, beside the classic hyper-parameterization of a MLP classifier, we tested the addition of a dropout layer before each densely connected layer. Dropout layers have been employed in predictive maintenance classifier given their effectiveness in preventing the network to overfit over noisy data [72,73]. The hyper-parameters space includes the activation function (*identity, logistic, tanh, relu*), the optimization algorithm (*lbfgs, sgd, adam*), the L2 regularization term (from 0.000001 to 0.001), the learning rate (*constant, inverse scaling, adaptive*), the maximum number of training iterations (from 500 to 4000), the dropout probability per layer (from 0 to 0.4), and the number of neurons per layer i.e. [32,32,32,4], [64,64,64], [264,64,16]. The best set of hyper-parameters for each case study have been employed to parametrize the classifier and measure its classification performances, as well as provide global interpretability to the model by using LIME.

Table 8
CIs of the F1-scores obtained without the feature selection module, with different optimization of the baseline MLP classifier and its improved version.

Feature learner	Classifier	CASE A	CASE B	CASE C	CASE D
AE	**MLP**	0.788 ± 0.048	0.831 ± 0.039	0.884 ± 0.036	0.861 ± 0.017
	Optimized	**0.909 ± 0.021**	**0.858 ± 0.075**	0.866 ± 0.011	**0.903 ± 0.021**
	Optimized + Dropout	0.902 ± 0.061	0.828 ± 0.069	**0.921 ± 0.048**	0.891 ± 0.019
VAE	**MLP**	0.935 ± 0.017	0.836 ± 0.058	0.860 ± 0.029	0.720 ± 0.0163
	Optimized	0.936 ± 0.021	**0.870 ± 0.023**	0.884 ± 0.051	0.787 ± 0.014
	Optimized + Dropout	**0.947 ± 0.017**	0.863 ± 0.039	**0.917 ± 0.063**	**0.797 ± 0.025**

5. Results

In this section, we show our experimental results. Each one of them is presented as a 95 % confidence interval (CI) obtained with a stratified Monte Carlo cross fold validation, i.e. by randomly picking 70 % of the data as training set and 30 % as testing set, with 10 repetitions. The hardware platform used for our experiments employs an *AMD EPYC CPU* (8 cores, 16 Threads, 2195 MHz), 23 Gigabyte RAM, and an *NVIDIA Tesla T4* GPU. Firstly, we assess the impact of different numbers of best encodings, the ones to be concatenated to generate the final array of features for the classifier. We test 4, 6, and 8 as numbers of best encodings, both with AE and VAE as feature learners. The feature learner has been trained by using an early stop configuration, i.e. the training stops if the error does not decrease significantly for 10 subsequent epochs. The classifier employs 500 training epochs. Table 2 shows the F1-scores confidence interval of the obtained with the three case studies.

Given a number of best encodings, AE always performs better than VAE. Among all, the case study C results to benefit more from a greater number of encodings. Thus, a number of best encodings equal to 8, is employed as configuration with all the next experimentations. We aim at assessing if the classifier is "accommodating" for the complexity of the classification, potentially hiding the poor quality of the features. In this case, lowering the number of training epochs of the classifier should result in a major decrease in the performances. In Table 3, we show the degradation of the average F1-scores obtained by lowering the training epochs of the classifier from 500 to 100. This results in minor performance decrease with each case study, both for the feature extractor based on VAE and AE, thus the classifier is not compensating for a poor-quality feature extraction.

In the majority of the trials of the first experiment, the training of the

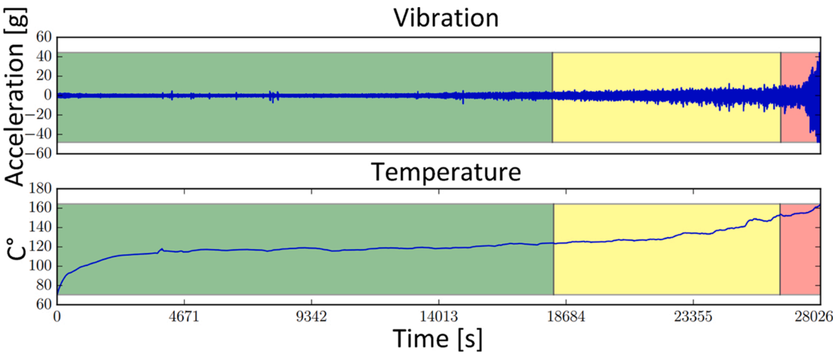


Fig. 4. Example of observations' labeling as regular (green), degraded (yellow), and critical (red) health stage. Case Study A (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

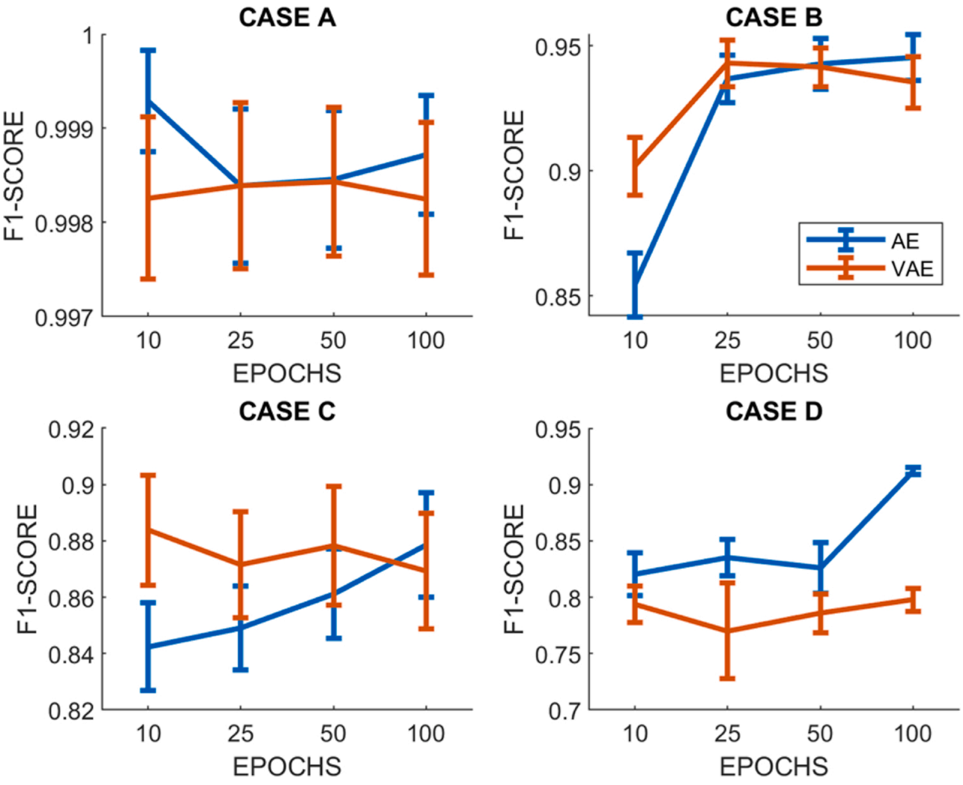


Fig. 5. CI of the F1-scores obtained by varying the number of training epochs of the feature learner.

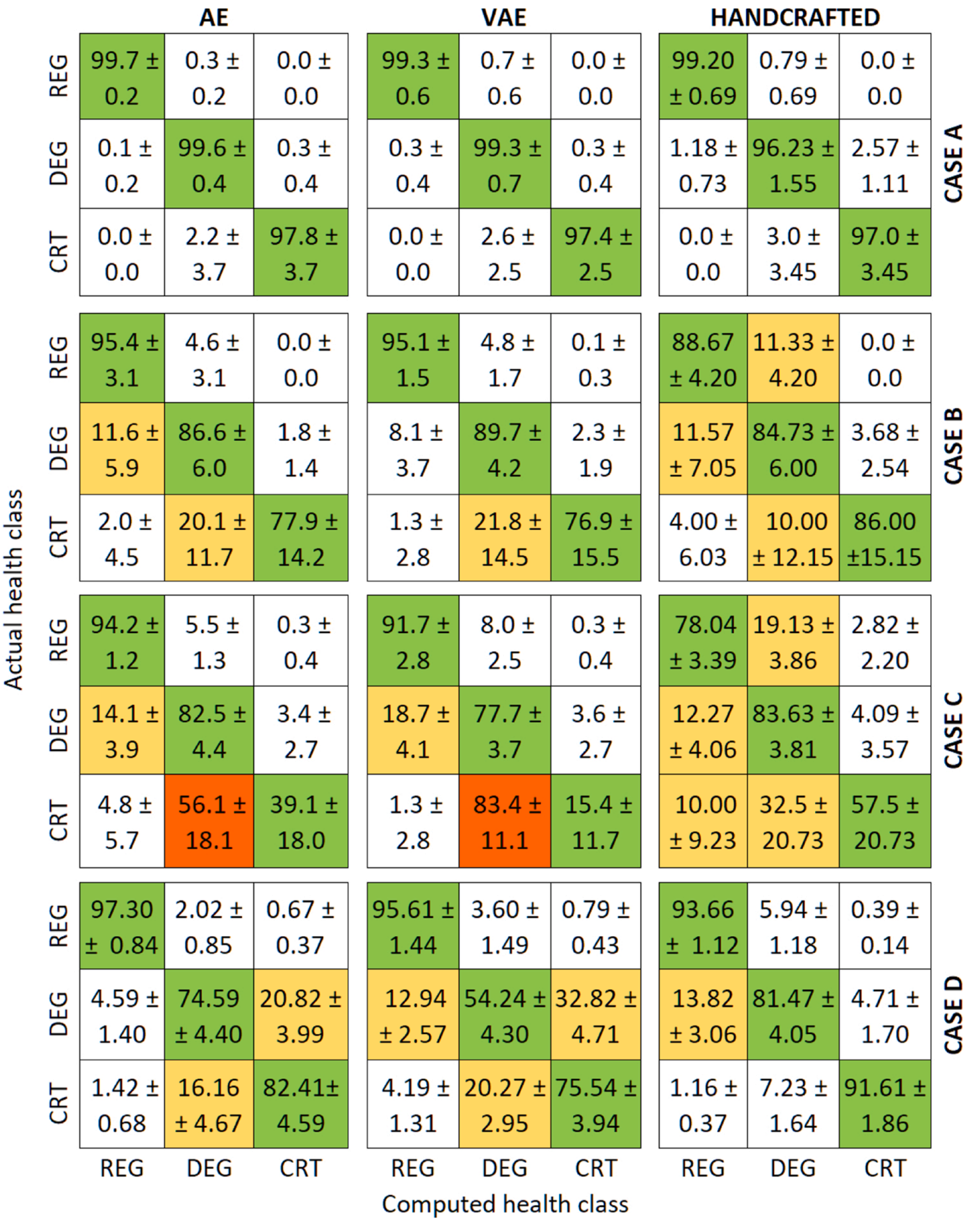
feature extractor converged with less than 100 epochs, and never exceeded 125 training epochs. To test how the performances are affected by the number of training epochs of the feature extractor, we collect the classification F1-scores with 10, 25, 50, and 100 training epochs. According to the results shown in Fig. 5, the F1-scores CIs partially overlap and do not allow to speculate beyond the trend of the average F1-scores. The only exception to this trend is the case study D. In this regard, the feature extractor based on AE seems to benefit more from a greater number of training epochs, and especially in the case studies B, C and D.

We compare the effectiveness of our approach with a classification based on a set of handcrafted features. Those are often used in bearing health stage classification and have been presented in the previous section. We also analyze how the performances vary by evaluating a single input signal rather than their combination. When employing the handcrafted features, we consider both input signals by concatenating the features obtained from each one of them. Moreover, we compare the proposed approaches with another one based on Long-Short Term Memory (LSTM) neural networks, due to their strong ability in modelling the temporal component of time series. Specifically, in [74] the authors extract 8 time-domain features and 2 similarity-based measures to train an LSTM-based neural network and derive the degradation assessment from the vibration signals of the Pronostia dataset. The obtained F1-scores are reported in Table 4. Our approach results in better performances with respect to those obtained with the other approaches. In addition, with the only exception of case study C, the performances obtained by considering both input signals are better or comparable than the one obtained by considering one single signal. Finally, the feature extractor based on AE always performs better than the one with VAE, except for the single temperature signal. This may suggest that, by being more stationary, this measure (i.e. the temperature) makes it easier to generate a set of encodings having a Gaussian distribution, which is a fundamental assumption of the feature learning mechanism based on VAE [75].

The problem addressed in this work is an unbalanced classification problem [76], and therefore it may require an additional effort while training the system, e.g. employing some oversampling approach to cope with imbalanced data [77]. In this regard, the autoencoder can be employed to generate features that easily separate classes even in the presence of unbalanced classes in the training set [78,79]. For example, authors in [80] and [81]conclude that a good feature learning approach allows a classifier to perform well even in case of class imbalance thanks to its ability to cluster similar instances in the latent space [82]. Indeed, with our approach the features learned via autoencoder are selected by clustering them and evaluating the match between the class labels and the clusters. Since the number of clusters corresponds to the number of degradation stages, the clustering procedure always generates 3 clusters. The match between the clusters and the classes' labels ensures that the features passed to the classifier are only those that place the instances of the same classes in proximity to each other and distant from other classes in the latent space. The features learned and selected in this way facilitate the classification, handling to a certain extent the unbalance issues in the training set. Of course, this is not true for the case of hand-crafted features. Thus, to evaluate the ability of the system to cope with the class imbalance in the dataset we decompose the classification performance according to each class, by presenting the resulting confusion matrices. With the handcrafted features, a balanced class weight is implemented during the classifier's training.

In Fig. 6, at the *j-th* row and column *k-th* of each confusion matrix, we show the confidence interval of the percentage of the samples of class *j* classified as class *k* [83]. The considered health stage classes are regular (REG), degraded (DEG), and critical (CRT). In green we highlight the percentage of correctly classified samples, in yellow the errors between 10 % and 50 %, in orange the errors greater than 50 %. According to the results shown in Fig. 6, our approach can manage the class unbalance, but this property can be affected by the complexity of the case study. Indeed, the case studies with lower classification performances correspond also to those in which the classification error on the minority classes is higher than the same obtained with the handcrafted features.

As said in Section 3, high-quality features are supposed to be close to each other if extracted from samples of the same class, distant otherwise.



	Actual health class	AE			VAE			HANDCRAFTED		
		REG	DEG	CRT	REG	DEG	CRT	REG	DEG	CRT
CASE A	REG	99.7 ± 0.2	0.3 ± 0.2	0.0 ± 0.0	99.3 ± 0.6	0.7 ± 0.6	0.0 ± 0.0	99.20 ± 0.69	0.79 ± 0.69	0.0 ± 0.0
	DEG	0.1 ± 0.2	99.6 ± 0.4	0.3 ± 0.4	0.3 ± 0.4	99.3 ± 0.7	0.3 ± 0.4	1.18 ± 0.73	96.23 ± 1.55	2.57 ± 1.11
	CRT	0.0 ± 0.0	2.2 ± 3.7	97.8 ± 3.7	0.0 ± 0.0	2.6 ± 2.5	97.4 ± 2.5	0.0 ± 0.0	3.0 ± 3.45	97.0 ± 3.45
CASE B	REG	95.4 ± 3.1	4.6 ± 3.1	0.0 ± 0.0	95.1 ± 1.5	4.8 ± 1.7	0.1 ± 0.3	88.67 ± 4.20	11.33 ± 4.20	0.0 ± 0.0
	DEG	11.6 ± 5.9	86.6 ± 6.0	1.8 ± 1.4	8.1 ± 3.7	89.7 ± 4.2	2.3 ± 1.9	11.57 ± 7.05	84.73 ± 6.00	3.68 ± 2.54
	CRT	2.0 ± 4.5	20.1 ± 11.7	77.9 ± 14.2	1.3 ± 2.8	21.8 ± 14.5	76.9 ± 15.5	4.00 ± 6.03	10.00 ± 12.15	86.00 ±15.15
CASE C	REG	94.2 ± 1.2	5.5 ± 1.3	0.3 ± 0.4	91.7 ± 2.8	8.0 ± 2.5	0.3 ± 0.4	78.04 ± 3.39	19.13 ± 3.86	2.82 ± 2.20
	DEG	14.1 ± 3.9	82.5 ± 4.4	3.4 ± 2.7	18.7 ± 4.1	77.7 ± 3.7	3.6 ± 2.7	12.27 ± 4.06	83.63 ± 3.81	4.09 ± 3.57
	CRT	4.8 ± 5.7	56.1 ± 18.1	39.1 ± 18.0	1.3 ± 2.8	83.4 ± 11.1	15.4 ± 11.7	10.00 ± 9.23	32.5 ± 20.73	57.5 ± 20.73
CASE D	REG	97.30 ± 0.84	2.02 ± 0.85	0.67 ± 0.37	95.61 ± 1.44	3.60 ± 1.49	0.79 ± 0.43	93.66 ± 1.12	5.94 ± 1.18	0.39 ± 0.14
	DEG	4.59 ± 1.40	74.59 ± 4.40	20.82 ± 3.99	12.94 ± 2.57	54.24 ± 4.30	32.82 ± 4.71	13.82 ± 3.06	81.47 ± 4.05	4.71 ± 1.70
	CRT	1.42 ± 0.68	16.16 ± 4.67	82.41± 4.59	4.19 ± 1.31	20.27 ± 2.95	75.54 ± 3.94	1.16 ± 0.37	7.23 ± 1.64	91.61 ± 1.86

Computed health class

Fig. 6. Confusion matrix with 10 repetitions. At the *j-th* row and column *k-th*, we show the confidence interval of the percentage of samples of class *j* classified as class *k*. The possible health classes are regular (REG), degraded (DEG), and critical (CRT).

To have further insight on the quality of the features generated, we process them with an unsupervised clustering approach (i.e., K-means with 3 clusters) and evaluate if the resulting clustering labels match the arrangement of the actual health stage labels. The results are shown in Table 5. The best results per case study are highlighted in bold. Those are obtained with the feature extractor based on AE in three case studies out of four.

It is possible to assume that the noise in a real-world degradation process is a random white Gaussian noise [76]. To test how this affects the recognition performance of the VAE-based approach, white Gaussian noise has been added to the vibration signal of each case study. The white Gaussian noise has been parametrized according to its signal-noise ratio (SNR). SNR values of 0.7, 0.8, 0.9, and 1 are tested, and the obtained results are shown in Table 6. With SNR equal to 1 the approach based on VAE loses from 1% to 10 % of its performance and maintains almost the same values up to a ratio between signal and noise power of 0.7. This trend supports the one found in [84].

The training time is a management cost that must be considered when employing a machine-learning approach. In Table 7, we show the duration (in seconds) of a training epoch for the feature extractor (both with VAE and AE) and the classifier. The feature extractor based on AE offers quicker training epochs. The durations of a training epoch with case study A and D are larger due to the higher number of examples to process (Table 1). The classifier has a clearly smaller duration of a training epoch.

Finally, we evaluate the capability of the feature selection module of (i) filtering out noisy or non-informative learned features for classification, and (ii) generating a global explanation of the proposed method. We repeated the measurement of the classification performances without the feature selection module, as explained in Section 4. As shown in Table 8, by using the proposed MLP as a classifier, the recognition performances drop at least by 5%. Even with the best-improved configuration of the classifier, the performances do not improve significantly nor consistently across the case studies confirming how complex the management of noisy features is without the feature selection module.

We further investigate the capability of the feature selection module to explain the model outcome and prove its reliability with respect to

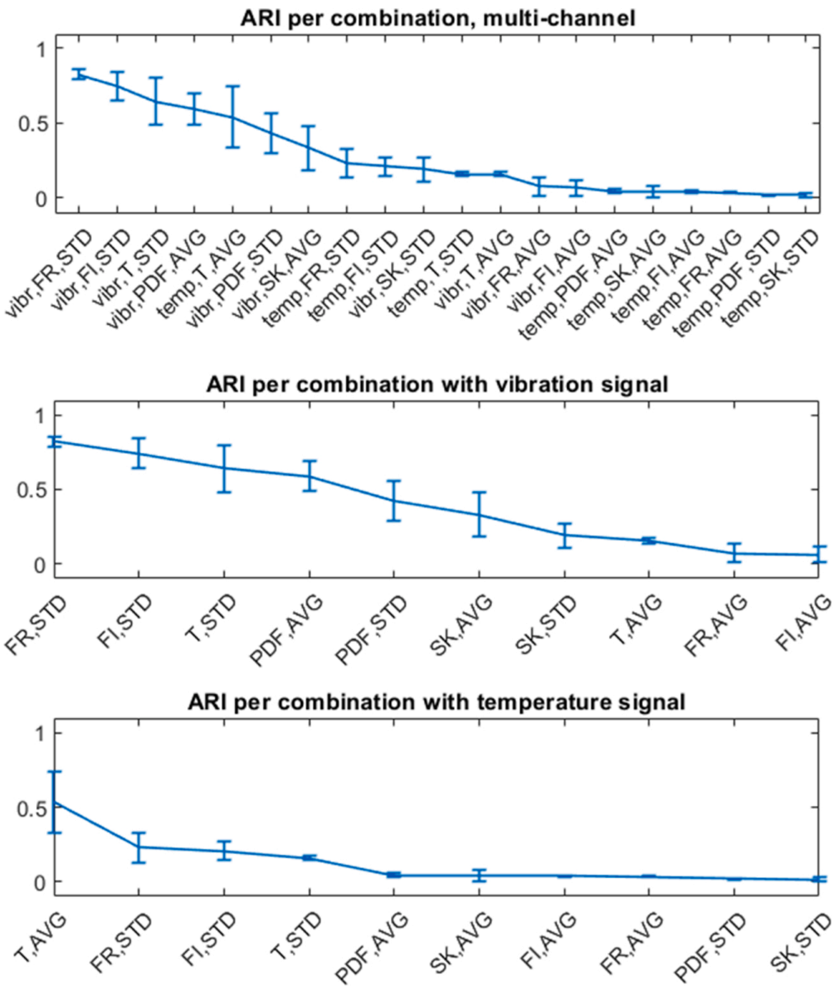


Fig. 7. 95 % CIs of the adjusted rand index used to assess the importance of each combination (input signal and data preparation) in the model for case study A.

common knowledge. As an example, in Fig. 7 we show the distribution of the resulting adjusted rand index (ARI) with case study A; the higher ARI, the more a combination is considered informative. The combinations are distinguished according to their (i) input signal: vibration (*vibr*) and temperature (*temp*); (ii) domain transformation: time-domain (*T*), real part of the Fourier transform (*FR*), the imaginary part of Fourier transform (*FI*), probability density function (*PDF*), and spectral kurtosis (*SK*); and (iii) aggregation operator: mean (*AVG*), or standard deviation (*STD*). Clearly, the vibration is considered more informative for this classification task, in fact among the 8 combinations with the highest ARI, 6 are obtained from the vibration signal. Indeed, compared to the vibration, the temperature seems to have a less consistent trend within the same health stage (Fig. 4). Moreover, by observing how the combinations are distributed according to the input signal (Fig. 7), the two best combinations for the vibration signals are based on frequency domain transformations, whereas the most important combination for the temperature employs the time domain.

To compare the capability of the feature selection module to generate model's global explanations (i.e. the rank of input signals and domain transformation), we consider the best AE-based approach among the ones shown in Table 8 for case study A and employ it in conjunction with LIME with the submodular pick algorithm to provide global interpretability to the model. Since the number of samples needed to generate a comprehensive global explanation of the model is not known a priori, we use a number of instances equal to 1%, 3%, and 10 % of their total amount. The contribution of each feature in the model has been averaged by the input signal and domain transformation. By using both 1%, 3%, and 10 % of instances to generate the explanation, the real (FR) and imaginary (FI) components of the Fourier transform of the temperature signal are always placed among the first 4 most important features. Since the temperature signal exhibits a non-oscillatory behavior, the accountability of this global explanation may be questioned, especially if the ranking obtained with LIME (Fig. 8) is compared to the one obtained with our approach (Fig. 7).

6. Conclusions

In this work, we propose a predictive maintenance approach based on a feature learning mechanism, aimed at replacing manual feature engineering. In contrast with most of the feature learning approaches

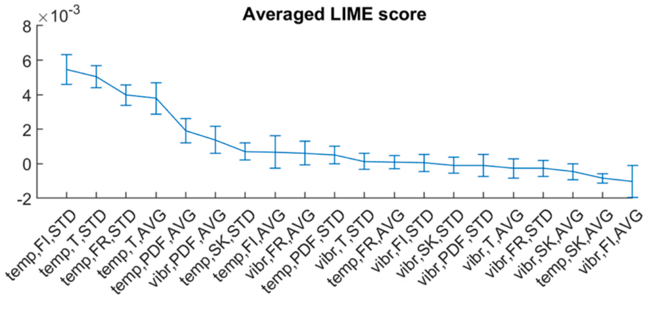


Fig. 8. Rank obtained with LIME via submodular pick algorithm, by averaging the contribution of each feature for each input signal and domain transformation. 10 % of the total amount of instances have been used to generate the global explanation. Case study A.

based on deep learning, our approach is able to extract informative features from different time series while providing useful insights about the most informative combinations of time series (e.g. vibration or temperature) and data transformation (e.g. frequency or time domain). We tested our approach with a real-world dataset consisting of the temperature and vibration run-to-failure time series of bearings. The obtained results show that our approach (i) is effective, resulting in an average F1-score not lower than 0.94 in all case studies, (ii) offers better or comparable performances with respect to the ones obtained with manual feature engineering or with a feature extractor based on VAE, (iii) results in low costs for its management and training, (iv) provides high-quality features, i.e. close to each other within the same health class and different from each other otherwise, and (v) allows to interpret the learned model and assess its reliability with respect to common knowledge, e.g. which data preparation is preferable according to the measure analyzed, and which measure is supposed to be more informative for a given degradation process. On the other hand, our approach results in lower performances with the case studies characterized by a less progressive degradation process, i.e., in which a single health stage sporadically presents behaviors that are typical of a more severe degradation stage. This happens especially with case study C and D. This shortcoming can be addressed with an approach that directly constrains the feature learning to the classification, i.e., with approaches based on disentangled data representations [85]. For this reason, we aim at introducing this technology in future developments of the proposed approach. Moreover, given that the autoencoder has shown some potential to cope with unbalanced classification problems, in future work we intend to explore more in-depth this aspect. It is indeed possible to modify the internal structure of the autoencoder to improve the separation of classes in the latency space regardless of the unbalanced distribution of instances among the classes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by (i) the company Fabio Perini S.p. A - Körber Tissue in the project “Intelligent Data Analysis algorithms for monitoring mono and multi-agent systems: Digital Applications in the Tissue Industry”, and by (ii) the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence). The authors thank Amedeo Pochiero for his work on the subject during his master thesis.

References

[1] O’Donovan P, Leahy K, Bruton K, O’Sullivan DTJ. An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. J Big Data 2015;2:25.

[2] Alfeo AL, Appio FP, Cimino MGCA, Lazzeri A, Martini A, Vaglini G. An adaptive stigmergy-based system for evaluating technological indicator dynamics in the context of smart specialization. Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods 2016:497–502.

[3] Alfeo AL, Cimino MGCA, Manco G, Ritacco E, Vaglini G. Using an autoencoder in the design of an anomaly detector for smart manufacturing. Pattern Recognit Lett 2020;136:272–8. https://doi.org/10.1016/j.patrec.2020.06.008.

[4] Ran Y, Zhou X, Lin P, Wen Y, Deng R. A survey of predictive maintenance: systems, purposes and approaches. ArXiv Preprint ArXiv:191207383 2019.

[5] Jimenez JJM, Schwartz S, Vingerhoeds R, Grabot B, Salaün M. Towards multi-model approaches to predictive maintenance: a systematic literature survey on diagnostics and prognostics. J Manuf Syst 2020;56:539–57.

[6] Wan J, Tang S, Li D, Wang S, Liu C, Abbas H, et al. A manufacturing big data solution for active preventive maintenance. IEEE Trans Industr Inform 2017;13:2039–47.

[7] Lei Y, Li N, Guo L, Li N, Yan T, et al. Machinery health prognostics: a systematic review from data acquisition to RUL prediction. Mech Syst Signal Process 2018;104:799–834.

[8] Chen L, Xu G, Zhang S, Yan W, Wu Q. Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks. J Manuf Syst 2020;54:1–11.

[9] Yu W, Dillon T, Mostafa F, Rahayu W, Liu Y. A global manufacturing big data ecosystem for fault detection in predictive maintenance. IEEE Trans Industr Inform 2019;16:183–92.

[10] Nguyen KTP, Medjaher K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliability Eng System Safety 2019;188:251–62.

[11] Scanlon P, Kavanagh DF, Boland FM. Residual life prediction of rotating machines using acoustic noise signals. IEEE Trans Instrum Meas 2012;62:95–108.

[12] Kimotho JK, Sondermann-Wölke C, Meyer T, Sextro W. Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution–Particle swarm optimization. Chem Eng Trans 2013:33.

[13] Fink O. Data-driven intelligent predictive maintenance of industrial assets. Women in Industrial and Systems Engineering. Springer; 2020. p. 589–605.

[14] Schölkopf B, Smola A, Müller K-R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 1998;10:1299–319.

[15] Yan W, Yu L. On accurate and reliable anomaly detection for gas turbine combustors: a deep learning approach. ArXiv Preprint ArXiv:190809238 2019.

[16] Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35:1798–828.

[17] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A, Bottou L. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010:11.

[18] Zhong G, Ling X, Wang L-N. From shallow feature learning to deep learning: benefits from the width and depth of deep architectures. Wiley Interdiscip Rev Data Min Knowl Discov 2019;9:e1255.

[19] Charte D, Charte F, del Jesus MJ, Herrera F. An analysis on the use of autoencoders for representation learning: fundamentals, learning task case studies, explainability and challenges. Neurocomputing 2020;404:93–107.

[20] Yan X, Liu Y, Jia M. Health condition identification for rolling bearing using a multi-domain indicator-based optimized stacked denoising autoencoder. Struct Health Monit 2020;19:1602–26.

[21] Qi Y, Shen C, Liu J, Li X, Li D, Zhu Z. An automatic feature learning and fault diagnosis method based on stacked sparse autoencoder. International Workshop of Advanced Manufacturing and Automation 2017:367–75.

[22] Nguyen A, Yosinski J, Clune J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015:427–36.

[23] Kraus M, Feuerriegel S. Forecasting remaining useful life: interpretable deep learning approach via variational Bayesian inferences. Decis Support Syst 2019;125:113100.

[24] Zhang W, Yang D, Wang H. Data-driven methods for predictive maintenance of industrial equipment: a survey. IEEE Syst J 2019;13:2213–27.

[25] Tang S, Yuan S, Zhu Y. Deep learning-based intelligent fault diagnosis methods toward rotating machinery. IEEE Access 2019;8:9335–46.

[26] Guyon I, Gunn S, Nikravesh M, Zadeh LA. Feature extraction: foundations and applications, vol. 207. Springer; 2008.

[27] Pearson III KL. On lines and planes of closest fit to systems of points in space. London, Edinburgh, and Dublin Philosophical Magazine J Sci. 1901;2:559–72.

[28] Joliffe IT, Morgan BJT. Principal component analysis and exploratory factor analysis. Stat Methods Med Res 1992;1:69–95.

[29] Fisher RA. The use of multiple measurements in taxonomic problems. Ann Eugen 1936;7:179–88.

[30] Schölkopf B. Statistical learning and kernel methods. Data Fusion and Perception. Springer; 2001. p. 3–24.

[31] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning, vol. 1. MIT press Cambridge; 2016.

[32] Dai J, Wang J, Huang W, Shi J, Zhu Z. Machinery health monitoring based on unsupervised feature learning via generative adversarial networks. IEEE/ASME Trans Mechatron 2020;25:2252–63.

[33] Oja E. Data compression, feature extraction, and autoassociation in feedforward neural networks. Artificial Neural Networks 1991.

[34] Pulgar FJ, Charte F, Rivera AJ, del Jesus MJ. Choosing the proper autoencoder for feature fusion based on data complexity and classifiers: analysis, tips and guidelines. Inf Fusion 2020;54:44–60.

[35] Shao H, Jiang H, Wang F, Zhao H. An enhancement deep feature fusion method for rotating machinery fault diagnosis. Knowledge Based Syst 2017;119:200–20.

[36] Shao H, Jiang H, Zhao H, Wang F. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. Mech Syst Signal Process 2017;95:187–204.

[37] Lu C, Wang Z-Y, Qin W-L, Ma J. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. Signal Processing 2017;130:377–88.

[38] Verma NK, Gupta VK, Sharma M, Sevakula RK. Intelligent condition based monitoring of rotating machines using sparse auto-encoders. 2013 IEEE Conference on Prognostics and Health Management (PHM) 2013:1–7.

[39] Sun W, Shao S, Zhao R, Yan R, Zhang X, Chen X. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. Measurement 2016;89:171–8.

[40] Ren L, Sun Y, Cui J, Zhang L. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. J Manuf Syst 2018;48:71–7.

[41] Jiang G, He H, Xie P, Tang Y. Stacked multilevel-denoising autoencoders: a new representation learning approach for wind turbine gearbox fault diagnosis. IEEE Trans Instrum Meas 2017;66:2391–402.

[42] Ping G, Chen J, Pan T, Pan J. Degradation feature extraction using multi-source monitoring data via logarithmic normal distribution based variational autoencoder. Comput Ind 2019;109:72–82.
[43] Zhai S, Gehring B, Reinhart G. Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning. J Manuf Syst 2021.
[44] Chen Z, Li W. Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. IEEE Trans Instrum Meas 2017;66:1693–702.
[45] Charte D, Charte F, Garcia S, del Jesus MJ, Herrera F. A practical tutorial on autoencoders for nonlinear feature fusion: taxonomy, models, software and guidelines. Inf Fusion 2018;44:78–96.
[46] Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L. Explaining explanations: an overview of interpretability of machine learning. 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA) 2018:80–9.
[47] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM Computing Surveys (CSUR) 2018;51:1–42.
[48] Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. European Conference on Computer Vision 2014:818–33.
[49] Ribeiro MT, Singh S, Guestrin C. “why should i trust you?” Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016:1135–44.
[50] Lundberg S, Lee S-I. A unified approach to interpreting model predictions. ArXiv Preprint ArXiv:170507874 2017.
[51] Alvarez-Melis D, Jaakkola TS. On the robustness of interpretability methods. ArXiv Preprint ArXiv:180608049 2018.
[52] ElShawi R, Sherif Y, Al-Mallah M, Sakr S. ILIME: local and global interpretable model-agnostic explainer of Black-Box decision. European Conference on Advances in Databases and Information Systems 2019:53–68.
[53] van der Linden I, Haned H, Kanoulas E. Global aggregations of local explanations for black box models. SIGIR’ 19: The 42nd International ACM SIGIR Conference on Research & Development in Information Retrieval 2019.
[54] Locatello F, Bauer S, Lucic M, Raetsch G, Gelly S, Schölkopf B, et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. International Conference on Machine Learning. 2019. p. 4114–24.
[55] Arrieta AB, Diaz-Rodriguez N, del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 2020;58:82–115.
[56] Lee N, Azarian MH, Pecht MG. An explainable deep learning-based prognostic model for rotating machinery. ArXiv Preprint ArXiv:200413608 2020.
[57] Feng Z, Yu Z, Yang Y, Jing Y, Jiang J, Song M. Interpretable partitioned embedding for customized multi-item fashion outfit composition. Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval 2018:143–51.
[58] Pandarakone SE, Masuko M, Mizuno Y, Nakamura H. Deep neural network based bearing fault diagnosis of induction motor using fast fourier transform analysis. 2018 IEEE Energy Conversion Congress and Exposition (ECCE) 2018:3214–21.
[59] Sadoughi M, Downey A, Bunge G, Ranawat A, Hu C, Laflamme S. A deep learning-based approach for fault diagnosis of rolling element bearings. Annual Conference of the PHM Society 2018;10.
[60] Amruthnath N, Gupta T. A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. 2018 5th International Conference on Industrial Engineering and Applications (ICIEA) 2018:355–61.
[61] Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: is a correction for chance necessary? Proceedings of the 26th Annual International Conference on Machine Learning 2009:1073–80.
[62] Kasahara T, Yonezawa Y, Ueda Y, Nambo H. Assessing machine condition using MLP and VAE-based classifiers using acceleration sensor data. International Conference on Management Science and Engineering Management 2019:581–91.
[63] Nectoux P, Gouriveau R, Medjaher K, Ramasso E, Chebel-Morello B, Zerhouni N, et al. PRONOSTIA: an experimental platform for bearings accelerated degradation tests. In: IEEE International Conference on Prognostics and Health Management; 2012. p. 1–8.
[64] Bechhoefer E, van Hecke B, He D. Processing for improved spectral analysis. Annual Conference of the Prognostics and Health Management Society 2013:14–7.
[65] Sandwell G. Basic understanding of machinery vibration. VIBES Corp.; 2020. n.d.
[66] Patterson D. Vibration analysis of motors in the service center. EASA; 2007. n.d.
[67] Mao W, He J, Tang J, Li Y. Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network. Adv Mech Eng 2018;10:1687814018817184.
[68] Zhu A, Meng Y, Zhang C. An improved Adam algorithm using look-ahead. Proceedings of the 2017 International Conference on Deep Learning Technologies 2017:19–22.
[69] Hou X, Shen L, Sun K, Qiu G. Deep feature consistent variational autoencoder. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV) 2017:1133–41.
[70] Hamadache M, Jung JH, Park J, Youn BD. A comprehensive review of artificial intelligence-based approaches for rolling element bearing PHM: shallow and deep learning. JMST Advances 2019;1:125–51.
[71] Wu J, Chen X-Y, Zhang H, Xiong L-D, Lei H, Deng S-H. Hyperparameter optimization for machine learning models based on Bayesian optimization. J Electr Sci Technol 2019;17:26–40.
[72] Li X, Kong X, Liu Z, Hu Z, Shi C. A novel framework for early pitting fault diagnosis of rotating machinery based on dilated CNN combined with spatial dropout. IEEE Access 2021;9:29243–52.
[73] Piotrowski AP, Napiorkowski JJ, Piotrowska AE. Impact of deep learning-based dropout on shallow neural networks applied to stream temperature modelling. Earth Sci Rev 2020;201:103076.
[74] Farahat A, Gupta C, et al. Similarity-based feature extraction from vibration data for prognostics. Annual Conference of the PHM Society 2020;12:10.
[75] Spinner T, Körner J, Görtler J, Deussen O. Towards an interpretable latent space: an intuitive comparison of autoencoders with variational autoencoders. IEEE VIS 2018 2018.
[76] Lee J, Lee YC, Kim JT. Fault detection based on one-class deep learning for manufacturing applications limited to an imbalanced database. J Manuf Syst 2020;57:357–66.
[77] Zhang Y, Li X, Gao L, Wang L, Wen L. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. J Manuf Syst 2018;48:34–50.
[78] Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from class-imbalanced data: review of methods and applications. Expert Syst Appl 2017;73:220–39.
[79] Fernández A, Garcia S, Galar M, Prati RC, Krawczyk B, Herrera F. Learning from imbalanced data sets, vol. 11. Springer; 2018.
[80] Ng WWY, Zeng G, Zhang J, Yeung DS, Pedrycz W. Dual autoencoders features for imbalance classification problem. Pattern Recognit 2016;60:875–89.
[81] Yeh Y-C, Hsu C-Y. Application of auto-encoder for time series classification with class imbalance. Proceedings of the Asia Pacific Industrial Engineering & Management Science Conference 2019:14–7.
[82] Zamini M, Montazer G. Credit card fraud detection using autoencoder based clustering. 2018 9th International Symposium on Telecommunications (IST) 2018:486–91.
[83] Xu H, Xu D, Chen S, Ma W, Shi Z. Rapid determination of soil class based on visible-near infrared, mid-infrared spectroscopy and data fusion. Remote Sens (Basel) 2020;12:1512.
[84] Ellefsen AL, Bjørlykhaug E, Æsøy V, Zhang H. An unsupervised reconstruction-based fault detection algorithm for maritime components. IEEE Access 2019;7:16101–9.
[85] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. ArXiv Preprint ArXiv:160603657 2016.

Antonio Luca Alfeo is a Postdoc research fellow at the Department of Information Engineering of the University of Pisa. In 2019, he received the pH.D. degree from the pH.D. Program in Smart Computing with a thesis addressing the design of bioinspired approaches for machine learning and data analysis. In 2018 he was a visiting pH.D. student at the Media Lab of the Massachusetts Institute of Technology. His research interests address the design of machine learning pipelines with applications in the contexts of smart mobility, e-health, and industry 4.0.

Mario G.C.A. Cimino is with the Department of Information Engineering (University of Pisa) as an Associate Professor. He is also a research associate at the Institute for Informatics and Telematics (IIT) of the Italian National Research Agency (CNR). In 2006, he was a visiting pH.D. student at the University of Alberta, Canada. In 2007, he received the pH.D. degree in Information Engineering from the University of Pisa. His research focus lies in the areas of Swarm Intelligence and Business/Social Process Analysis, with particular emphasis on Stigmergic Computing, Workflow Mining and Simulation. He is (co-) author of more than 70 publications.

Gigliola Vaglini received the M.S. degree in Computer Science from the University of Pisa. She was a research assistant at the University of Pisa, Department of Computer Science, an Associate Professor at the University of Naples, Federico II, Department of Mathematics, and from 2002 she is a Full Professor at the University of Pisa, Department of Information Engineering. Her research addressed formal methods for specification and verification of concurrent and distributed systems; in particular, she worked on model checking. More recently her research activity focused on the stigmergic paradigm aimed at achieving distributed control on autonomous systems.