Reliability Engineering and System Safety 224 (2022) 108482

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Health indicator for machine condition monitoring built in the latent space of a deep autoencoder

Ana González-Muñiz *, Ignacio Díaz, Abel A. Cuadrado, Diego García-Pérez

Electrical Engineering Department, University of Oviedo, Edif. Departmental Oeste 2, Campus de Viesques s/n, Gijón 33204, Spain

ARTICLE INFO

Keywords:
Health indicator
Deep autoencoder
Latent space
Anomaly detection
Engineering systems

ABSTRACT

The construction of effective health indicators plays a key role in the engineering systems field: they reflect the degradation degree of the system under study, thus providing vital information for critical tasks ranging from anomaly detection to remaining useful life estimation, with benefits such as reduced maintenance costs, improved productivity or increased machine availability. The reconstruction error of deep autoencoders has been widely used in the literature for this purpose, but this approach does not fully exploit the hierarchical nature of deep models. Instead, we propose to take advantage of the disentangled representations of data that are available in the latent space of autoencoders, by using the latent reconstruction error as machine health indicator. We have tested our proposal on three different datasets, considering two types of autoencoders (deep autoencoder and variational autoencoder), and comparing its performance with that of state-of-the-art approaches in terms of well-known quality metrics. The results of the research demonstrate the capability of our health indicator to outperform conventional approaches, in the three datasets, and regardless of the type of autoencoder used to generate the residuals. In addition, we provide some intuition on the suitability of latent spaces for the monitoring of machinery condition.

1. Introduction

In the last decade there has been a growing interest in the monitoring of machinery condition [1–3]. Machines are critical equipment in a wide variety of domains, and it has become crucial to optimize and guarantee safety in their operation, searching for higher productivity and process efficiency, and benefits such as reduced operating costs, longer machine life or improved operating uptime [4].

Condition monitoring systems in the literature have traditionally been focused on knowledge-based [5] or physical-based approaches [6]. However, with the rapid development of smart instruments, digital communication networks and computing techniques, there is nowadays an increasing availability of condition monitoring data, which has brought research attention to data-driven methods for the monitoring of machine health [7].

In the study of these methods, special efforts have been dedicated to the construction of health indicators (HIs) [8–10]. HIs reflect the degradation degree of the system under study, providing valuable input information for tasks such as fault diagnosis, anomaly detection (AD) or remaining useful life (RUL) estimation. For example, in the field of AD, the state of a system is usually declared anomalous when its HI (or HIs) meets certain conditions such as exceeding a predetermined threshold [11,12]. In prognostic tasks, such as RUL estimation, HIs are also used in many ways: similarity methods estimate the RUL of machines by comparison of degradation profiles that are built on one or more HIs [13]; other approaches propose to build degradation models, based on HI data, to predict the future condition of the machines [14], etc. The performance of these tasks heavily depends on the quality of the health indicator [15,16], so the construction of effective HIs plays a key role in the context of machine health monitoring.

In traditional knowledge-based or physical-based approaches, HIs are typically manually built, requiring expertise or prior knowledge of the system that can sometimes be difficult or even impossible to obtain. Also, these HIs are often designed for a specific degradation process, and it can be tough to make them generalize well to others. Hence, constructing a good HI is a challenging work that in data-driven methods does not rely on the human, but on powerful machine learning techniques able to extract HIs automatically from process data, also known as *feature learning* [17].

As a branch of machine learning, deep learning models have set in recent years an exciting new trend with a big impact on big data analysis [18], showing promising results in fields such as image or voice recognition [19,20], but also in system health management [21].

* Corresponding author.
E-mail addresses: anaglezmuniz@gmail.com (A. González-Muñiz), idiaz@uniovi.es (I. Díaz), cuadrado@isa.uniovi.es (A.A. Cuadrado), diegogarcia@isa.uniovi.es (D. García-Pérez).

https://doi.org/10.1016/j.ress.2022.108482
Received 15 June 2021; Received in revised form 9 February 2022; Accepted 20 March 2022
Available online 2 April 2022

They are built by stacking multiple processing layers, which conform compositional hierarchies in which the higher level features are the composition of the lower level ones, what gives deep models the ability to find the best representations of data, known as *representation learning* [22]. Thanks to this hierarchical nature, they have become powerful feature learning tools and therefore have attracted many researchers in the field of HI construction [23–25].

Authors in the literature have used different types of deep techniques for the construction of HIs, being the most popular: convolutional neural networks (CNNs) [26,27], recurrent neural networks (RNNs) [15,28], Generative Adversarial Networks (GANs) [29] and deep autoencoders (AEs) [30,31], on which we have focused our research. Deep autoencoders are feed-forward multi-layer neural networks in which the desired output is the input itself: they compress the input data into a lower-dimensional projection and then reconstruct the output data from this representation. Due to their deep nature, they have the ability to reduce the dimension of the input in a hierarchical way, leading to high quality reconstructions of data [32,33]. In the context of health monitoring, deep autoencoders are usually trained on normal (healthy) data, so they learn to successfully reconstruct normal samples, but fail to reproduce anomalous ones, leading to a high reconstruction error. Hence, reconstruction error has been widely used in the literature as health indicator [34–37].

However, recent studies have indicated that reconstruction-based methods are limited by measuring the reconstruction quality only in the input space, as they do not take advantage of the disentangled representations of data that are available in the hidden spaces of the autoencoder. For example, in [38] authors propose a novelty detection method named *reconstruction along projection pathway* (RaPP), which computes the hidden reconstruction error: RaPP compares the hidden activations of the input with those of the reconstructed input, and proves to outperform the traditional reconstruction error-based approach.

In view of this situation, we propose to explore the potential of reconstruction error as health indicator, but in a low dimensional space: the latent space of the autoencoder. We propose to use the latent reconstruction error as machine health indicator and evaluate its performance for prognostics tasks. In this way, we follow the philosophy of the RaPP approach, since we aim to exploit the information contained in the hidden spaces of the autoencoder, but we focus only on the latent space information. Latent spaces have proved to successfully capture the underlying structure in data, providing us with low-dimensional, compact and meaningful representations of the original data [39]. Therefore, latent reconstruction error would result in a disentangled representation of reconstruction error, that we hypothesize might be a more robust and, consequently, more suitable health indicator for prognostics tasks than the reconstruction error itself.

Our proposal consists of a deep autoencoder, which has been trained to reconstruct samples of healthy behaviour, and a health indicator, being the latent reconstruction error of the autoencoder. We have tested this proposal on three different datasets and we have compared its performance with that of state-of-the-art approaches in terms of monotonicity, trendability and prognosability. The results of the research demonstrate that our proposal achieves better performance than its alternative approaches (reconstruction error and RaPP approach). We also provide some intuition on the suitability of latent spaces for machine health monitoring. In summary, the contributions of this research are as follows: (1) we propose a variation of the RaPP approach that results in a novel deep autoencoder-based health indicator for machine health monitoring; (2) we provide a geometric interpretation of the proposal that illustrates the consistency of the health indicator in the presence of machine degradation.

The rest of this document is organized as follows. In Section 2, we introduce the theoretical foundation of deep autoencoders and present different health indicators used in the literature. Section 3 describes the architecture of the proposed system. The datasets and the results of the proposal are presented in Section 4. Finally, the conclusions are set out in Section 5.

2. Related literature

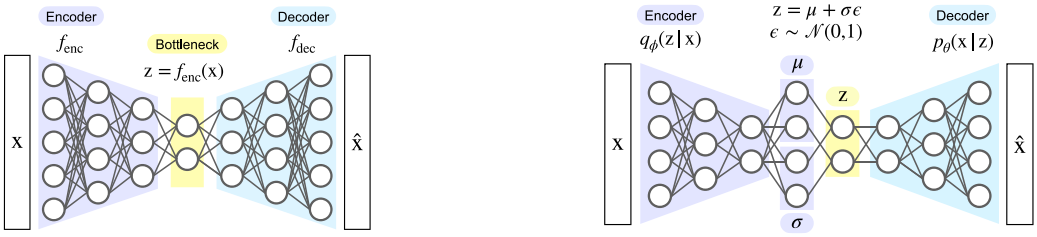
There has been extensive research in the literature concerning machine health monitoring: we can find several approaches in the state of the art, depending on the application domain and even the dataset [7,40]. In the typical scenario, authors deal with highly class-imbalanced datasets containing mostly normal behaviour data and few, if any, examples of anomalous behaviour, what makes traditional discriminative learning schemes unsuitable for health monitoring [41]. The common alternative is to construct a model of normal behaviour (based on available normal data), so that condition monitoring would depend on how well new data fit this model: it is a residual error-based approach in which residuals are used as indicators of machinery health condition. As detailed below, deep autoencoders have been successful in this domain, showing great skills in the modelling of normal behaviour, and providing residuals that are accurate health indicators.

2.1. Deep autoencoders

With the proliferation in recent years of deep learning techniques, several authors have used the reconstruction error of deep autoencoders as health indicator [11,36,37,42]. As we see in Fig. 1, the architecture of autoencoders consists of an encoder f_{enc}, which outputs a latent representation $\mathbf{z}$ of the input data $\mathbf{x}$, and a decoder f_{dec} that reconstructs the input data ($\hat{\mathbf{x}}$) from its latent representation $\mathbf{z}$. In between, there is a bottleneck (typically one or more low-dimensional layers), so the identity map is not a possible solution and the model is forced to learn the underlying low-dimensional structure in data. During the learning process, the architecture is trained using the gradient descent method [43] by means of backpropagation [44] in order to minimize the difference between $\mathbf{x}$ and $\hat{\mathbf{x}}$. If the autoencoder is trained with normal operation data, the resulting model will be able to reconstruct incoming normal samples with a small reconstruction error, but its ability to reconstruct novelty samples will be limited, as they do not resemble the data used for training. Consequently, high reconstruction errors are associated with system deviations from normal behaviour, such as machine degradation.

We have also considered the use of variational autoencoders (VAEs) [45] in our study. VAEs emerged as a probability-based extension of deep autoencoders and have been gaining popularity in the last years, showing promising results as generative models with applications in several fields, such as audio, text, or image generation [46–48]. In contrast to traditional autoencoders, VAEs impose restrictions on the distribution of latent variables and, in doing so, learn the probability density function of training data. As we can see in Fig. 1, the VAE consists of an encoder $q_\phi(\mathbf{z}|\mathbf{x})$, being an approximate posterior, and a decoder $p_\theta(\mathbf{x}|\mathbf{z})$, being the likelihood of the data $\mathbf{x}$ given the latent variable $\mathbf{z}$. According to this scheme, the encoder becomes a variational inference network, mapping input data to (approximate) posterior distributions over the latent space, and the decoder operates as a generative network, mapping arbitrary latent coordinates back to distributions over the original data space. To achieve this, it is assumed that input data can be sampled from a unit Gaussian distribution of latent parameters, so that during the learning process the model is trained by simultaneously optimizing two loss functions: a reconstruction loss (as in conventional autoencoders); and the Kullback–Leibler divergence D_{KL} between the learned latent distribution and a prior unit Gaussian. So we can understand the VAE as a deep autoencoder with an additional regularization provided by the D_{KL} term. In this way, we obtain an autoencoder that works as a generative model.

VAEs are therefore a powerful extension of deep autoencoders and are expected to play a significant role in the future of health monitoring algorithms [21]. Not surprisingly, they have already outperformed ordinary autoencoders in reconstruction error-based novelty detection [49]; and their latent spaces have proven to be powerful feature extractors, providing compact representations of the data that have been successfully used in fault detection applications [50,51].



(a) Deep autoencoder.

(b) Deep variational autoencoder.

Fig. 1. Autoencoder architecture.

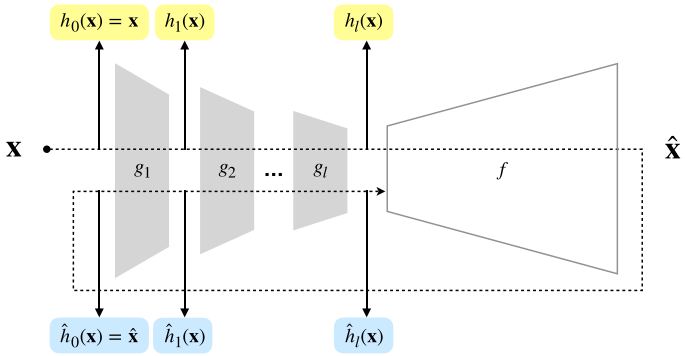


Fig. 2. RaPP approach.
Source: Figure adapted from [38].

2.2. Reconstruction error as health indicator

As mentioned before, when autoencoders are trained on normal data, the novelty or degradation degree of an incoming sample is typically measured by means of its reconstruction error (1), which compares the input sample $\mathbf{x}$ with its reconstruction $\hat{\mathbf{x}}$. In this context, samples with large residual errors are considered more likely to be novel [41], as they are far from the manifold described by the trained autoencoder.

$$\varepsilon_{REC}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 \tag{1}$$

Although this approach has proven successful in the literature [11,36,37,42], it does not really take advantage of one of the most notable proven strengths of deep models, which is their ability to learn hierarchical representations of data [22]. Instead, the reconstruction along projection pathway (RaPP) approach, recently presented in [38], proposes to exploit such hierarchical nature by extending the reconstruction error to the hidden spaces of the autoencoder. Rather than comparing $\mathbf{x}$ and $\hat{\mathbf{x}}$ only in the input space, as the ordinary approach does, RaPP proposes to project the input and its reconstruction into the hidden spaces, thus obtaining pairs of activation values that will be later aggregated to quantify the novelty of the input.

More formally, let $A = f \circ g$ be the trained autoencoder (where f is the decoder and g is the encoder), and l be the number of layers in g. Then the partial computation of g would be defined as: $g_{:i} = g_i \circ \cdots \circ g_1$, for $1 \leq i \leq l$. Accordingly, when we feed $\mathbf{x}$ and $\hat{\mathbf{x}}$ into A, we obtain pairs $(h_i, \hat{h}_i)$ of their hidden representations, where $h_i(\mathbf{x}) = g_{:i}(\mathbf{x})$ and $\hat{h}_i(\mathbf{x}) = g_{:i}(\hat{\mathbf{x}})$. This procedure is illustrated in Fig. 2.

For the aggregation of the activations, authors present two metrics, resulting in two different health indicators (ε_{SAP} and ε_{NAP}) derived from the RaPP approach. The simple aggregation along pathway (SAP) for a sample $\mathbf{x}$ is defined by summing the square of Euclidean distances for all pairs of activations:

$$\varepsilon_{SAP}(\mathbf{x}) = \|h(\mathbf{x}) - \hat{h}(\mathbf{x})\|_2 \tag{2}$$

where $h(\mathbf{x}) = [h_0(\mathbf{x}), \ldots, h_l(\mathbf{x})]$ and $\hat{h}(\mathbf{x}) = [\hat{h}_0(\mathbf{x}), \ldots, \hat{h}_l(\mathbf{x})]$.

The normalized aggregation along pathway (NAP) is an extension of SAP which proposes to normalize the distances before their aggregation, in order to alleviate the dependency between hidden layers. To be precise: let distances be $d(\mathbf{x}) = h(\mathbf{x}) - \hat{h}(\mathbf{x})$; given a training set $\mathbf{X}$, let $\mathbf{D}$ be a matrix whose ith row corresponds to $d(\mathbf{x}_i)$ for $\mathbf{x}_i \in \mathbf{X}$; and let $\bar{\mathbf{D}}$ be the column-wise centred matrix of $\mathbf{D}$. In this context, ε_{NAP} for a given sample $\mathbf{x}$ is defined as follows:

$$\varepsilon_{NAP}(\mathbf{x}) = \|(d(\mathbf{x}) - \mu_{\mathbf{X}})^\top \mathbf{V}\boldsymbol{\Sigma}^{-1}\|_2 \tag{3}$$

where $d(\mathbf{x})$ is expressed as a column vector, $\mu_{\mathbf{X}}$ is the column-wise mean of $\mathbf{D}$, $\boldsymbol{\Sigma}$ is a diagonal square matrix containing the singular values of $\bar{\mathbf{D}}$, and $\mathbf{V}$ is a matrix containing the right singular vectors of $\bar{\mathbf{D}}$. Note that ε_{NAP} is based on the concept of Mahalanobis distance.

Experiments in the literature [38,52] have shown that these RaPP indicators are able to outperform the traditional reconstruction error-based approach. Therefore, the information provided by the hidden spaces of the autoencoder has proven to be of great value, and seems to have great potential in the construction of health indicators.

3. Proposed method

Following the philosophy of the RaPP approach, we build our health indicator based on the disentangled representations of data provided by the hidden spaces of the autoencoder. But we focus only on the latent space information: we propose to use the latent reconstruction error as machine health indicator. The motivation for this approach and its mathematical formulation are presented below.

3.1. Motivation

Generally, high-dimensional data is considered to lie in a low-dimensional space [53,54], which means that, in most situations, high-dimensional phenomena can actually be dominated by a small amount of variables. Therefore, a suitable low-dimensional manifold, such as the latent space of an autoencoder, would reveal the underlying sources of variation of the phenomena under study.

As explained in Section 2.1, deep autoencoders are trained to project the input data into a low-dimensional latent space and then reconstruct the output data from this compressed representation. Thanks to their hierarchical nature, they have shown a great ability to learn meaningful representations of data and their latent spaces have become powerful feature extractors [39,55]. Latent spaces capture the underlying structure in data, providing us with compact projections that are more robust to undesired properties (such as the presence of noise or artefacts in data) than high-dimensional spaces [56,57].

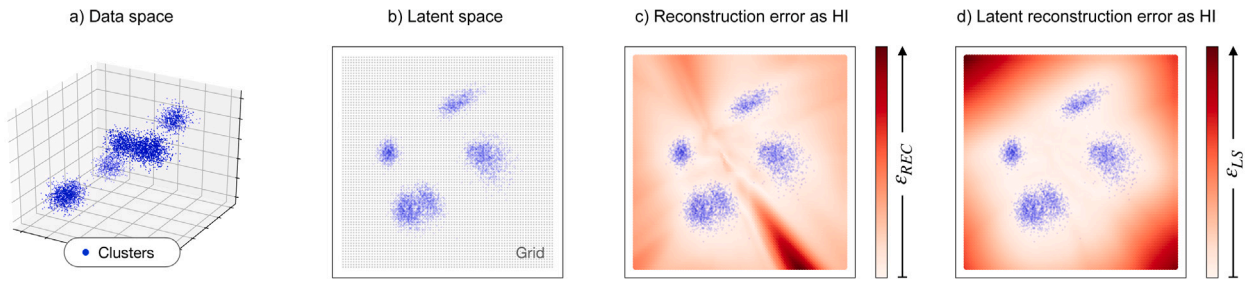


Fig. 3. Latent reconstruction error for a set of clusters.

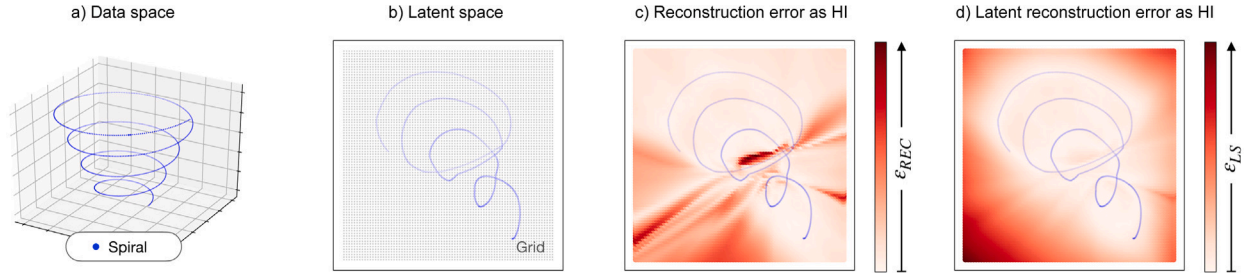


Fig. 4. Latent reconstruction error for a spiral.

Therefore, we propose to compute the reconstruction error in this latent feature space rather than in the original data space. The latent reconstruction error will thus result in a disentangled representation of the reconstruction error, that we hypothesize might be a more robust and, consequently, more suitable health indicator for prognostics tasks than the reconstruction error itself. This can be seen in the figures above, where we illustrate the comparison with two different three-dimensional (3D) examples: a set of clusters and a spiral (Figs. 3 and 4, respectively). In both cases, we trained an autoencoder to reconstruct the 3D data samples (considering they are representative of normal behaviour) using a 2D latent space, and then created a grid for which we computed the reconstruction error ε_{REC} and the latent reconstruction error ε_{LS}, which are represented by a colour scale going from white (low error, normal behaviour) to red (high error, faulty behaviour), so that they reflect the deviation degree from process normal behaviour. In Fig. 3.c and Fig. 4.c we can see that ε_{REC} is sensitive to multiple sources of variation, while ε_{LS} (Fig. 3.d, Fig. 4.d) is only sensitive to deviations from normal behaviour (it is low in high-density areas and gradually increases towards low-density areas), being an apparently more consistent health indicator.

It should also be noted that this proposal is a simplification of the RaPP approach: instead of computing the reconstruction error in all hidden spaces, we put the focus on the hidden space that provides the most compact and parsimonious representation of the data, summarizing in a reduced number of factors the main modes of variation of the process, which is the latent space of the autoencoder. In Section 4, we will evaluate the performance of our proposal, compared to the RaPP approach and to the traditional reconstruction error-based solution, in the presence of a certain deviation from the process normal behaviour, being machine degradation.

3.2. Latent reconstruction error

The construction of our health indicator starts by training an autoencoder (consisting of an encoder g and a decoder f) only with normal data samples. Then, when we receive an incoming sample whose health we want to evaluate, we project the sample $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}}$ into the latent space of the autoencoder (Fig. 5), producing a pair of activation values $[h_l(\mathbf{x}), \hat{h}_l(\mathbf{x})]$ that will be aggregated

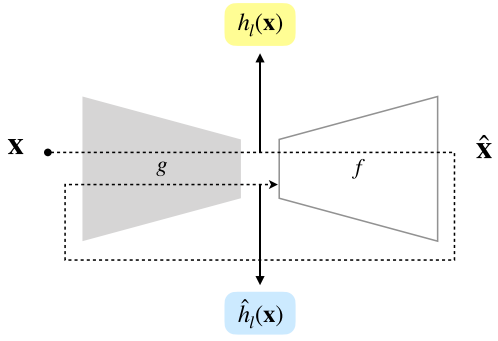


Fig. 5. Proposed method: we constrain the RaPP approach to the latent space.

to obtain the latent reconstruction error. For the aggregation, we use the same metrics as the RaPP approach (SAP, NAP), but constrained to the latent space, so we obtain two different health indicators: $\varepsilon_{SAP_{LS}}$ and $\varepsilon_{NAP_{LS}}$.

We define the latent reconstruction error $\varepsilon_{SAP_{LS}}$ (for the sake of clarity, in the previous subsection we used the term ε_{LS}) of a sample $\mathbf{x}$ as the Euclidean distance between its latent representation $h_l(\mathbf{x})$ and the latent representation of its reconstruction $\hat{h}_l(\mathbf{x})$:

$$\varepsilon_{SAP_{LS}}(\mathbf{x}) = \|h_l(\mathbf{x}) - \hat{h}_l(\mathbf{x})\|_2 \tag{4}$$

In the same way as in the RaPP approach, we can also normalize the distances before their aggregation. To be precise, we can compute $\varepsilon_{NAP_{LS}}$ according to Eq. (3), but in this case with distances $d(\mathbf{x}) = h_l(\mathbf{x}) - \hat{h}_l(\mathbf{x})$.

Hence, our proposal yields two health indicators ($\varepsilon_{SAP_{LS}}$, $\varepsilon_{NAP_{LS}}$), whose performance will be presented in the next section.

4. Results

We have tested our autoencoder-based health indicator on three different datasets, considering two types of autoencoders, and comparing its performance with that of state-of-the-art approaches. In the

Table 1
Datasets used in this research.

Dataset	Sensors	Runs	Cycles	Size	Train data	Test data
FD001	21	200	33727	33727 × 21	23692 × 21	10035 × 21
FD003	21	519	41316	41316 × 21	31289 × 21	10027 × 21
Mill	6	16	167	8350 × 600	3000 × 600	5350 × 600

following subsections we describe the datasets, the architecture of the autoencoders and the results of performance in terms of monotonicity, trendability and prognosability. We also visualize in this section the latent space of the trained autoencoders, bringing light on the potential of latent residuals for the monitoring of machinery degradation.

4.1. Datasets

The data used in this research has been taken from three publicly available datasets, which are presented in Table 1. Datasets FD001 and FD003 are part of the C-MAPSS repository [58], developed by NASA, which provides simulated degradation data of turbofan engines with different operating conditions and fault patterns. The Mill dataset [59], provided by the BEST lab at UC Berkeley, contains tool wear data from a milling machine running under various operating conditions.

More in detail, the FD001 dataset contains readings of 21 sensors for 200 runs (100 engines operating till end-of-life and 100 engines pruned some time prior to failure), where each run has a different duration, resulting in a dataset with a total of 33727 machine cycles. For each cycle, there is some additional information available (operational settings of the engines) that we have not included in our experiments. The FD003 dataset, which collects operating data from another fleet of engines, has been constructed in the same way: it consists of 519 runs (260 engines operating till end-of-life and 259 engines pruned some time prior to failure), resulting in a dataset of 41316 samples. Regarding data preprocessing, we have normalized both datasets by means of min–max scaling with range [0,1] (in detail, we have trained a scaler on the train data and then used it to transform both train and test data).

The Mill dataset contains readings of 6 sensors for 16 run-to-failure cases of different duration, recording a total of 167 machine cycles. In this dataset, each cycle of the machine is defined by 6 snapshot sequences (one per sensor) of size 9000 points, that we have preprocessed as follows: we have discarded the first and last 2000 points (corresponding to the turning on and turning off of the milling machine) obtaining vectors of size 5000 points, which we have split into 50 windows of 100 elements (with no overlapping). Concatenating the windows of all sensors, we obtain a total of 8350 samples of size 600 elements. Additional information about the machine cycle (such as operational settings or the type of material) has not been included in our experiments. Finally, as in the previous scenarios, we have normalized the dataset by means of min–max scaling with range [0,1] (we have trained a scaler on the train data and then used it to transform both train and test data).

Regarding the training and testing of our proposal, we have partitioned the datasets into two sets (train data and test data), as indicated in Table 1. In detail, we have considered the initial cycles of the machines to be representative of process normal behaviour and, therefore, they have been used to compose the training sets. In particular, for the Mill dataset we have considered as healthy samples those corresponding to the first four cycles of machine operation; for datasets FD001 and FD003, we have chosen as normal samples those with RUL greater than 80 cycles (in these datasets the RUL of each sample is a known variable). These normal samples have been used to build a model of process normal behaviour (Section 4.2); and the remaining samples have been used as test data in order to assess the performance of our health indicator (Section 4.3).

4.2. Model of normal behaviour

Our health indicator is built on the residuals of a deep autoencoder that is trained to reconstruct normal data samples, so that residuals of incoming samples become a measure of their deviation from normal behaviour. For this purpose, we have trained the autoencoders detailed below (Table 2, Table 3). It should be noted that we have used two types of architectures (deep autoencoder and variational autoencoder), since we aim to evaluate the performance of our HI considering different architectures for the generation of the residuals.

We have trained three deep autoencoders, one per dataset, using the minibatch gradient descent [43] and the Adam optimizer [60]. The number of epochs, the batch size, and the architectures of these models are shown in Table 2, and we have used the rectified linear unit (ReLU) function [61] as activation function in all their layers (except in the output and bottleneck layers, where we have used the identity function).

We have also trained three variational autoencoders, one per dataset, using the minibatch gradient descent [43] and the rmsprop optimizer [62]. The number of epochs and the batch size are indicated in Table 3. The architectures of these models (number of layers and number of neurons per layer) are also shown in the table, and we have used the ReLU function [61] as activation function in all their layers (except in the output and bottleneck layers, where we have used sigmoidal and identity functions, respectively).

It should also be noted that the tuning of all the parameters mentioned in this subsection (Table 2, Table 3) is based on the testing of different configurations. The chosen configurations are the ones that yielded the best results in terms of reconstruction error (another option could be to rely on the latent reconstruction error, but we have chosen the reconstruction error in order to be consistent with the loss function of the autoencoder). For the implementation of the architectures, we have used the Keras framework (with Tensorflow backend), which is an open source neural network library, widely used by the research community [63].

Finally, it is worth mentioning that our proposal is based on the assumption that the training datasets are representative of the full range of expected operating conditions. However, in systems with changing environments, normal conditions may change over time, potentially affecting the performance of our proposal. In such cases, a re-training of the autoencoder —triggered e.g. by a drift detector [64]— may be required in order to model the actual process normal behaviour.

4.3. Health indicator evaluation

In order to evaluate the performance of our proposal, we have computed the latent reconstruction error for the test set of each engineering system (FD001, FD003, Mill) based on the residuals of different architectures (deep autoencoder, variational autoencoder), and compared its performance with that of state-of-the-art approaches. In detail, we have considered three approaches for the construction of HIs: the traditional approach ε_{REC}, in which residuals are computed in the input space of the autoencoder; the RaPP approach, in which residuals are computed in the hidden spaces of the autoencoder (in accordance with the literature, we have considered the simple ε_{SAP} and normalized ε_{NAP} aggregation of residuals); and our approach, which proposes to compute the residuals in the latent space of the autoencoder (in the same way, we present two possible indicators, being $\varepsilon_{SAP_{LS}}$ and $\varepsilon_{NAP_{LS}}$). In order to evaluate the performance of these approaches, we have used three well-known quality metrics [65]: *monotonicity* (measure of the trend of the HI as the system evolves toward failure), *trendability* (measure of similarity between the trajectories of the HI in multiple run-to-failure experiments) and *prognosability* (measure of the variability of the HI at failure relative to the range between its initial and final values). The obtained results are shown in Table 4, where we

Table 2
Deep autoencoder architecture for each dataset.

	Number of epochs	Batch size	Number of layers	Number of neurons in the layers		
				Encoder	Bottleneck	Decoder
FD001	200	800	9	(21,10,20,10)	(2)	(10,20,10,21)
FD003	200	800	9	(21,10,20,10)	(2)	(10,20,10,21)
Mill	300	1000	7	(600,200,100)	(10)	(100,200,600)

Table 3
Variational autoencoder architecture for each dataset.

	Number of epochs	Batch size	Number of layers	Number of neurons in the layers		
				Encoder	Bottleneck	Decoder
FD001	400	300	11	(21,10,20,10)	(2,2,2)	(10,20,10,21)
FD003	400	300	11	(21,10,20,10)	(2,2,2)	(10,20,10,21)
Mill	400	300	9	(600,200,100)	(10,10,10)	(100,200,600)

Table 4
Results of performance in terms of monotonicity –mono–, trendability –tren– and prognosability –prog– (best results for each metric and dataset are presented in bold font).

	FD001			FD003			Mill		
HI	Mono	Tren	Prog	Mono	Tren	Prog	Mono	Tren	Prog
$\varepsilon_{NAP_{LS}}$	**0.455**	$\mathbf{3.025 \times 10^{-04}}$	**0.991**	**0.413**	$\mathbf{2.980 \times 10^{-04}}$	0.904	**0.406**	5.318×10^{-02}	**0.392**
$\varepsilon_{SAP_{LS}}$	0.416	1.867×10^{-04}	**0.991**	0.401	4.529×10^{-05}	0.898	0.357	$\mathbf{2.320 \times 10^{-01}}$	0.326
ε_{NAP}	0.156	2.252×10^{-05}	0.979	0.179	3.310×10^{-05}	**1.000**	0.260	7.083×10^{-03}	0.043
ε_{SAP}	0.265	1.358×10^{-04}	0.909	0.187	3.562×10^{-05}	0.790	0.306	1.589×10^{-01}	0.366
ε_{REC}	0.199	1.859×10^{-17}	0.893	0.214	2.629×10^{-18}	0.803	0.302	1.060×10^{-01}	0.344

(a) Results based on deep autoencoder residuals.

	FD001			FD003			Mill		
HI	Mono	Tren	Prog	Mono	Tren	Prog	Mono	Tren	Prog
$\varepsilon_{NAP_{LS}}$	**0.557**	$\mathbf{5.535 \times 10^{-03}}$	0.963	**0.561**	1.914×10^{-03}	0.950	0.267	6.358×10^{-03}	**0.384**
$\varepsilon_{SAP_{LS}}$	0.546	2.857×10^{-04}	**0.965**	0.534	$\mathbf{2.048 \times 10^{-03}}$	0.926	**0.371**	$\mathbf{9.227 \times 10^{-02}}$	0.349
ε_{NAP}	0.409	3.303×10^{-04}	0.964	0.224	1.204×10^{-05}	**1.000**	0.209	1.074×10^{-03}	0.021
ε_{SAP}	0.459	8.019×10^{-04}	0.964	0.494	4.833×10^{-04}	0.938	0.312	4.071×10^{-03}	0.351
ε_{REC}	0.263	7.423×10^{-05}	0.913	0.314	8.646×10^{-19}	0.913	0.333	4.451×10^{-02}	0.305

(b) Results based on variational autoencoder residuals.

have also included, for ease of comparison, a bar chart for each column of the table.

According to Table 4, the HIs based on the latent reconstruction error ($\varepsilon_{SAP_{LS}}$, $\varepsilon_{NAP_{LS}}$) lead to better results than state-of-the-art approaches. This can be seen in the results of the three datasets, and for both deep models, except for the prognosability in the FD003 dataset. Therefore, our method has proven to outperform the approaches in the literature, in different contexts, and regardless of the type of autoencoder used to generate the residuals. We can also notice that the two indicators derived from our proposal present good performance, but especially the normalized aggregation $\varepsilon_{NAP_{LS}}$.

In addition, we have also considered five time-domain statistical descriptors commonly used in the literature as health indicators: skewness, kurtosis, Root Mean Square (RMS) value, crest factor and variance [66]. To compute these features we have chosen windows of size 20 samples for FD001 and FD003 datasets, and of 100 samples for the Mill dataset; we have computed these features for each sensor in the datasets and reported the best-case results (Table 5). We can see that, although these indicators achieve good results in some metrics (monotonicity in FD001 and FD003 datasets for the RMS value, and prognosability in the Mill dataset), in general terms, they are less competitive than the indicators automatically extracted by the deep learning models. In fact, although manually engineered features have proven successful in many applications, choosing the right features is not always an easy task, as it may require expertise, prior knowledge of the machine or a strong mathematical basis.

Finally, we present a comparison of the residual-based health indicators for four test trajectories corresponding to the FD001 dataset (Fig. 6), the FD003 dataset (Fig. 7) and the Mill dataset (Fig. 8). As we can see in these figures, the HIs built on the hidden reconstruction error (ε_{SAP}, ε_{NAP}) show better performance than the traditional ε_{REC} approach, which reports noisy degradation profiles. The ε_{NAP} indicator has achieved excellent results in the literature in anomaly detection applications, being superior to ε_{SAP}, but here, in the context of condition monitoring, it presents an undesired property: it shows an abrupt increase at the end of machine life that can be noticed in Fig. 6.a.3, Fig. 7.a.3, Fig. 8.a.3 and Fig. 8.b.3. This behaviour hinders early fault detection, which is a critical aspect in predictive maintenance applications, and may explain the modest results achieved by ε_{NAP} in Table 4. We can also see that the HIs derived from our proposal ($\varepsilon_{SAP_{LS}}$, $\varepsilon_{NAP_{LS}}$) report similar degradation profiles, sharing a notable particularity: residuals remain low for the first samples of the trajectory, when the machine has not yet started to degrade, and then present a gradual increase (representing gradual degradation) until machine failure. Hence, these HIs appear to be sensitive only to deviations from normal behaviour (which is consistent with the expectations discussed in Section 3.1), leading to more robust degradation profiles, and consequently to better performance results, than the state-of-the-art approaches. We can also notice that all these approaches provide degradation profiles that reveal the degradation degree of the system under study, but have no direct physical meaning, which is a typical aspect of machine learning-based health indicators.

Given the good performance of our proposal observed in the results, we can say that the degradation profiles provided by the proposed health indicator represent valuable condition information that could be used as input data in prognostic tasks, such as RUL estimation applications. These applications combine the HI with available run-to-failure sets in order to provide a RUL estimation. In this context,

Table 5
Results obtained using statistical features (best results for each metric and dataset are presented in bold font; results that outperform deep learning approaches are presented in underlined font).

	FD001			FD003			Mill		
HI	Mono	Tren	Prog	Mono	Tren	Prog	Mono	Tren	Prog
SKEWNESS	0.154	3.278×10^{-20}	0.416	0.196	4.791×10^{-20}	0.362	0.291	4.000×10^{-03}	<u>0.563</u>
KURTOSIS	0.117	9.828×10^{-19}	0.378	0.142	$\mathbf{1.325 \times 10^{-05}}$	0.339	0.292	$\mathbf{3.860 \times 10^{-02}}$	**0.758**
RMS	**0.777**	5.677×10^{-20}	**0.651**	**0.838**	3.807×10^{-17}	**0.693**	**0.369**	9.000×10^{-04}	<u>0.413</u>
CREST FACTOR	0.232	$\mathbf{5.799 \times 10^{-05}}$	0.358	0.264	4.562×10^{-06}	0.205	0.270	7.600×10^{-03}	<u>0.670</u>
VARIANCE	0.151	3.333×10^{-05}	0.595	0.199	2.790×10^{-18}	0.108	0.338	7.000×10^{-04}	<u>0.656</u>

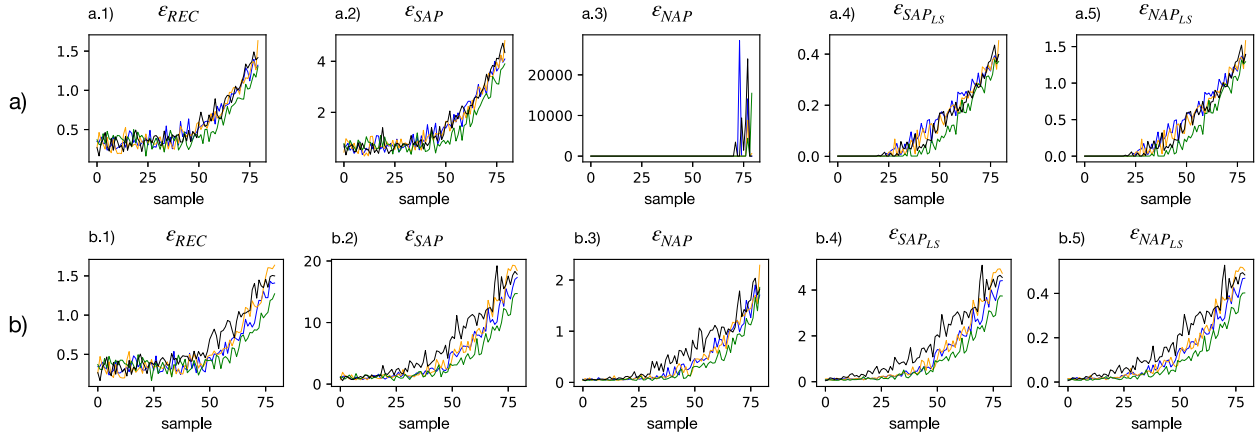


Fig. 6. HIs built on (a) deep autoencoder and (b) variational autoencoder residuals for four test degradation profiles belonging to dataset FD001.

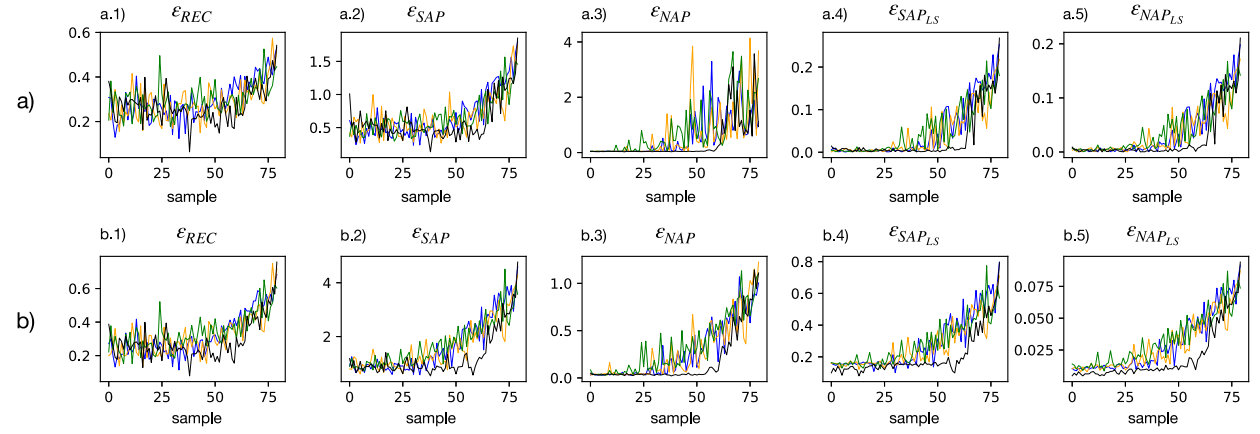


Fig. 7. HIs built on (a) deep autoencoder and (b) variational autoencoder residuals for four test degradation profiles belonging to dataset FD003.

similarity-based approaches propose to determine the RUL of the machine by means of comparison and matching of machine degradation profiles [13]; other approaches propose the use of degradation models, which identify the dynamics behind the degradation profiles to predict the future condition of the machine [67]; another popular strategy in the literature consists in using Support Vector Regressors (SVRs) to model the relationship between the health indicator and the RUL of the machine [68].

4.4. Geometric interpretation of our proposal

Proceeding in the same way as in Section 3.1, we can create a grid in the latent space of the autoencoder and use it as a degradation map of the monitored process. In particular, we analyse below the process degradation for the FD001 dataset, based on the deep autoencoder residuals.

We can see in Fig. 9 the latent space of the autoencoder, where we have displayed five degradation maps, each one built on a different health indicator. We have also projected the trajectories of four engines (the same trajectories shown in Fig. 6), including a vector indicating the direction of machine degradation and the probability density function of training data (in colour blue; and estimated by means of KDE —Kernel Density Estimation— with gaussian kernel).

This map comparison allows us to see how, in all cases, the HI increases in the direction of machine degradation, thus being consistent with the nature of the process. However, if we take a broad view of the maps, we can notice that each HI provides a complementary geometric interpretation of the latent space. For example, it is noticeable that the indicators derived from our proposal (Fig. 9.d, Fig. 9.e) yield maps with smoother and more regularized geometry than the state-of-the-art approaches (Fig. 9.a, Fig. 9.b, Fig. 9.c), and also better aligned with the density function of training data (which is representative of machine healthy behaviour). Therefore, the geometry of our proposal is more consistent with the degradation of the process, which explains the good results obtained by $\varepsilon_{SAP_{LS}}$ and $\varepsilon_{NAP_{LS}}$ in Table 4.

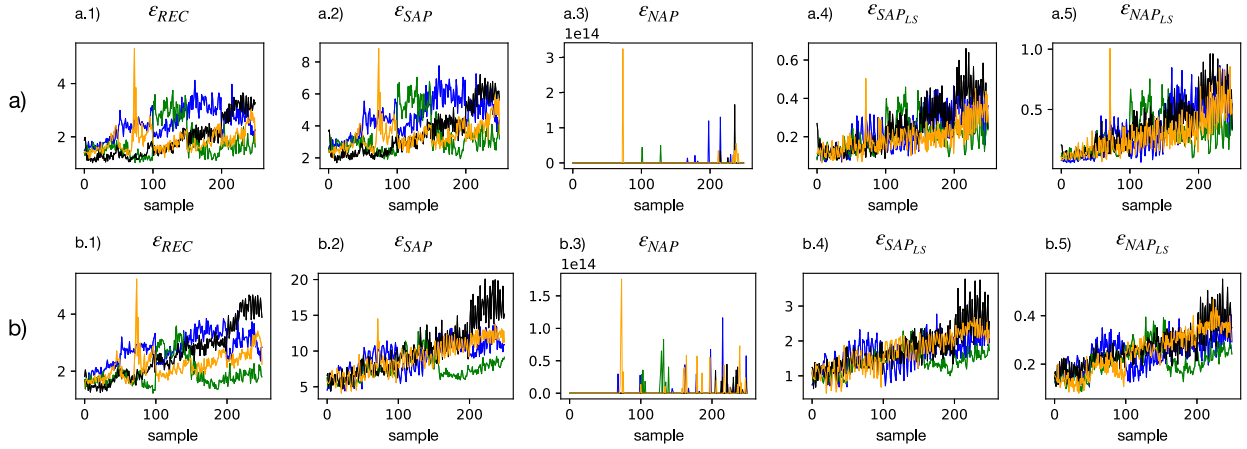


Fig. 8. HIs built on (a) deep autoencoder and (b) variational autoencoder residuals for four test degradation profiles belonging to the Mill dataset.

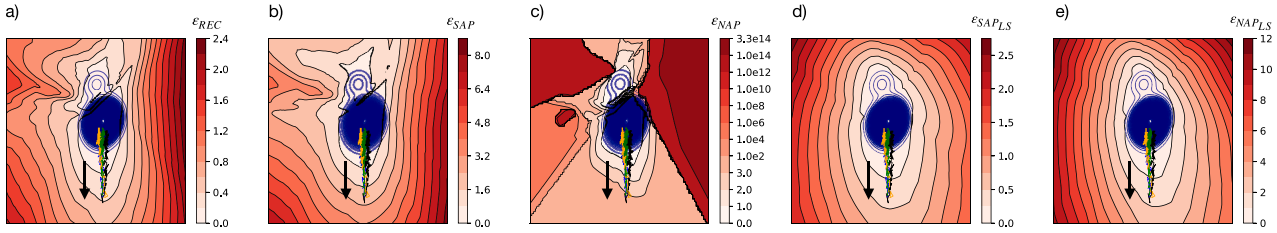


Fig. 9. Health indicator representation in the latent space of the deep autoencoder for the FD001 dataset: (a) ε_{REC}, (b) ε_{SAP}, (c) ε_{NAP}, (d) $\varepsilon_{SAP_{LS}}$, (e) $\varepsilon_{NAP_{LS}}$.

Finally, in view of these results, we can fully appreciate the information provided by the health indicators and how it may assist in the process of decision making in the engineering systems field. As shown in the previous section (Fig. 6, Fig. 7, Fig. 8), the magnitude of the health indicator reveals the degradation degree of the system under study, so we can interpret the indicator as a measure of machine degradation, thus providing valuable condition information that can be used as a decision-making aid in the planning of maintenance actions [69] or as input data in prognostic tasks such as RUL estimation [70]. Furthermore, the visualization of the machine degradation direction (Fig. 9) could reveal different types of degradation or anomalous behaviour, allowing us to identify which one the machine is suffering from and therefore to plan the most appropriate maintenance operations accordingly.

5. Conclusion

In this paper we have proposed the latent reconstruction error of deep autoencoders as machine health indicator. We have tested our proposal on three different datasets, considering two types of autoencoders (deep autoencoder and variational autoencoder), and comparing its performance with that of state-of-the-art approaches: the traditional reconstruction error approach, in which residuals are computed in the input space of the autoencoder; and the RaPP approach (recently presented in the literature), in which residuals are computed in the hidden spaces of the autoencoder. The results of the research (expressed in terms of monotonicity, trendability and prognosability) have proven the capability of our health indicator to outperform its competitors, in the three datasets, and regardless of the type of autoencoder used to generate the residuals.

Therefore, we could say that the quality of the health indicator is superior when the residuals are computed in the latent space of the autoencoder, rather than in hidden or input data spaces, as previously done in the literature. This suggests that, by providing compact and disentangled representations of the original data, latent spaces might also be revealing the underlying sources of variation —such as machine degradation— of the process under study. Hence, our research not only presents a novel health indicator, but also provides evidence of the potential of latent spaces for the monitoring of machinery condition.

Regarding future work, we are interested in exploring the applicability of our proposal in hybrid encoder–decoder architectures, such as deep adversarial autoencoders (AAEs). We are also interested in the semantic analysis of the latent space, so the study of the latent directions of machine degradation and their connection with the proposed health indicator will be the topic of our future work.

CRediT authorship contribution statement

Ana González-Muñiz: Conceptualization, Investigation, Software, Writing – Original Draft, Writing - Review & Editing. **Ignacio Díaz:** Supervision, Writing – Review & Editing. **Abel A. Cuadrado:** Writing – Review & Editing. **Diego García-Pérez:** Writing – Review & Editing.

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.

Acknowledgements

This work has been financed by the Spanish National Research Agency (under grant number PID2020-115401GB-I00/AEI/10.13039/501100011033). The authors would also like to thank the financial support provided by the Principado de Asturias government through the predoctoral grant “Severo Ochoa”.

References

[1] Jing L, Zhao M, Li P, Xu X. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. Measurement 2017;111:1–10.

[2] Stetco A, Dinmohammadi F, Zhao X, Robu V, Flynn D, Barnes M, et al. Machine learning methods for wind turbine condition monitoring: A review. Renew Energy 2019;133:620–35.

[3] Wakiru J, Pintelon L, Muchiri PN, Chemweno PK, Mburu S. Towards an innovative lubricant condition monitoring strategy for maintenance of ageing multi-unit systems.. Reliab Eng Syst Saf 2020;204:107200.

[4] Mobley RK. An introduction to predictive maintenance. Elsevier; 2002.

[5] Phillips P, Diston D. A knowledge driven approach to aerospace condition monitoring. Knowl-Based Syst 2011;24(6):915–27.

[6] An D, Kim NH, Choi J-H. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. Reliab Eng Syst Saf 2015;133:223–36.

[7] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. Mech Syst Signal Process 2019;115:213–37.

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

[9] Peng K, Jiao R, Dong J, Pi Y. A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter. Neurocomputing 2019;361:19–28.

[10] Ahmad W, Khan SA, Islam MM, Kim J-M. A reliable technique for remaining useful life estimation of rolling element bearings using dynamic regression models. Reliab Eng Syst Saf 2019;184:67–76.

[11] Martinelli M, Tronci E, Dipoppa G, Balducelli C. Electric power system anomaly detection using neural networks. In: International conference on knowledge-based and intelligent information and engineering systems. Springer; 2004, p. 1242–8.

[12] Zhou Y, Huang M, Chen Y, Tao Y. A novel health indicator for on-line lithium-ion batteries remaining useful life prediction. J Power Sources 2016;321:1–10.

[13] Yu W, Kim IY, Mechefske C. An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. Reliab Eng Syst Saf 2020;199:106926.

[14] Wen P, Zhao S, Chen S, Li Y. A generalized remaining useful life prediction method for complex systems based on composite health indicator. Reliab Eng Syst Saf 2021;205:107241.

[15] Guo L, Li N, Jia F, Lei Y, Lin J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. Neurocomputing 2017;240:98–109.

[16] Xiao L, Tang J, Zhang X, Bechhoefer E, Ding S. Remaining useful life prediction based on intentional noise injection and feature reconstruction. Reliab Eng Syst Saf 2021;107871.

[17] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.

[18] Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. J Big Data 2015;2(1):1–21.

[19] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal 2017;42:60–88.

[20] Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K. Speech recognition using deep neural networks: A systematic review. IEEE Access 2019;7:19143–65.

[21] Khan S, Yairi T. A review on the application of deep learning in system health management. Mech Syst Signal Process 2018;107:241–65.

[22] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35(8):1798–828.

[23] Guo L, Lei Y, Li N, Xing S. Deep convolution feature learning for health indicator construction of bearings. In: 2017 prognostics and system health management conference. IEEE; 2017, p. 1–6.

[24] 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.

[25] Zhang W, Li X, Ma H, Luo Z, Li X. Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions. Reliab Eng Syst Saf 2021;211:107556.

[26] Guo L, Lei Y, Li N, Yan T, Li N. Machinery health indicator construction based on convolutional neural networks considering trend burr. Neurocomputing 2018;292:142–50.

[27] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 2018;172:1–11.

[28] Xiang S, Qin Y, Luo J, Pu H, Tang B. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. Reliab Eng Syst Saf 2021;216:107927.

[29] Booyse W, Wilke DN, Heyns S. Deep digital twins for detection, diagnostics and prognostics. Mech Syst Signal Process 2020;140:106612.

[30] Xu F, Huang Z, Yang F, Wang D, Tsui KL. Constructing a health indicator for roller bearings by using a stacked auto-encoder with an exponential function to eliminate concussion. Appl Soft Comput 2020;89:106119.

[31] Chen D, Qin Y, Wang Y, Zhou J. Health indicator construction by quadratic function-based deep convolutional auto-encoder and its application into bearing RUL prediction. ISA Trans 2020.

[32] Tewari A, Zollhofer M, Kim H, Garrido P, Bernard F, Perez P, et al. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In: Proceedings of the IEEE international conference on computer vision workshops. 2017, p. 1274–83.

[33] Zhao F, Feng J, Zhao J, Yang W, Yan S. Robust lstm-autoencoders for face de-occlusion in the wild. IEEE Trans Image Process 2017;27(2):778–90.

[34] Malhotra P, TV V, Ramakrishnan A, Anand G, Vig L, Agarwal P, et al. Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder. 2016, arXiv preprint arXiv:1608.06154.

[35] Zhou C, Paffenroth RC. Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017, p. 665–74.

[36] Oh DY, Yun ID. Residual error based anomaly detection using auto-encoder in SMD machine sound. Sensors 2018;18(5):1308.

[37] Andresini G, Appice A, Di Mauro N, Loglisci C, Malerba D. Exploiting the auto-encoder residual error for intrusion detection. In: 2019 IEEE European symposium on security and privacy workshops. IEEE; 2019, p. 281–90.

[38] Kim KH, Shim S, Lim Y, Jeon J, Choi J, Kim B, et al. Rapp: Novelty detection with reconstruction along projection pathway. In: International conference on learning representations. 2019.

[39] Liu Y, Jun E, Li Q, Heer J. Latent space cartography: Visual analysis of vector space embeddings. In: Comput Graph Forum. 38, (3):Wiley Online Library; 2019, p. 67–78.

[40] Chalapathy R, Chawla S. Deep learning for anomaly detection: A survey. 2019, arXiv preprint arXiv:1901.03407.

[41] Pimentel MA, Clifton DA, Clifton L, Tarassenko L. A review of novelty detection. Signal Process 2014;99:215–49.

[42] Sakurada M, Yairi T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis. 2014, p. 4–11.

[43] Ruder S. An overview of gradient descent optimization algorithms. 2016, arXiv preprint arXiv:1609.04747.

[44] Rumelhart DE, Hinton GE, Williams RJ, et al. Learning representations by back-propagating errors. Cogn Model 1988;5(3):1.

[45] Kingma DP, Welling M. Auto-encoding variational bayes. 2013, arXiv preprint arXiv:1312.6114.

[46] Pu Y, Gan Z, Henao R, Yuan X, Li C, Stevens A, Carin L. Variational autoencoder for deep learning of images, labels and captions. 2016, arXiv preprint arXiv:1609.08976.

[47] Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. In: International conference on machine learning. PMLR; 2017, p. 1945–54.

[48] Roberts A, Engel J, Raffel C, Hawthorne C, Eck D. A hierarchical latent vector model for learning long-term structure in music. In: International conference on machine learning. PMLR; 2018, p. 4364–73.

[49] An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. Special Lect IE 2015;2(1):1–18.

[50] Michau G, Fink O. Domain adaptation for one-class classification: monitoring the health of critical systems under limited information. 2019, arXiv preprint arXiv:1907.09204.

[51] Michau G, Hu Y, Palmé T, Fink O. Feature learning for fault detection in high-dimensional condition monitoring signals. Proc Inst Mech Eng O J Risk Reliab 2020;234(1):104–15.

[52] Shin SY, Kim H-j. Extended autoencoder for novelty detection with reconstruction along projection pathway. Appl Sci 2020;10(13):4497.

[53] Hinton GE, Dayan P, Revow M. Modeling the manifolds of images of handwritten digits. IEEE Trans Neural Netw 1997;8(1):65–74.

[54] Seung HS, Lee DD. The manifold ways of perception. Science 2000;290(5500):2268–9.

[55] Yousefi-Azar M, Varadharajan V, Hamey L, Tupakula U. Autoencoder-based feature learning for cyber security applications. In: 2017 international joint conference on neural networks. (IJCNN), IEEE; 2017, p. 3854–61.

[56] Haufe S, Dähne S, Nikulin VV. Dimensionality reduction for the analysis of brain oscillations. NeuroImage 2014;101:583–97.

[57] Becker M, Lippel J, Stuhlsatz A, Zielke T. Robust dimensionality reduction for data visualization with deep neural networks. Graph Models 2020;108:101060.

[58] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 international conference on prognostics and health management. IEEE; 2008, p. 1–9.

[59] Agogino A, Goebel K. Milling data set. BEST lab, UC Berkeley; 2007.

[60] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.

[61] Hahnloser RH, Sarpeshkar R, Mahowald MA, Douglas RJ, Seung HS. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 2000;405(6789):947.

[62] Tieleman T, Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw Mach Learn 2012;4(2):26–31.
[63] Chollet F, et al. Keras. 2015, https://keras.io.
[64] Yang Z, Al-Dahidi S, Baraldi P, Zio E, Montelatici L. A novel concept drift detection method for incremental learning in nonstationary environments. IEEE Trans Neural Netw Learn Syst 2019;31(1):309–20.
[65] Coble JB. Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters. 2010.
[66] Yang H, Mathew J, Ma L. Vibration feature extraction techniques for fault diagnosis of rotating machinery: a literature survey. In: Asia-pacific vibration conference, no. 42460. 2003, p. 801–7.
[67] Park P, Jung M, Di Marco P. Remaining useful life estimation of bearings using data-driven ridge regression. Appl Sci 2020;10(24):8977.
[68] Khelif R, Chebel-Morello B, Malinowski S, Laajili E, Fnaiech F, Zerhouni N. Direct remaining useful life estimation based on support vector regression. IEEE Trans Ind Electron 2016;64(3):2276–85.
[69] Laloix T, Iung B, Voisin A, Romagne E. Parameter identification of health indicator aggregation for decision-making in predictive maintenance: Application to machine tool. CIRP Ann 2019;68(1):483–6.
[70] Zhao Z, Liang B, Wang X, Lu W. Remaining useful life prediction of aircraft engine based on degradation pattern learning. Reliab Eng Syst Saf 2017;164:74–83.