Applied Intelligence (2021) 52:10934–10964
https://doi.org/10.1007/s10489-021-03004-y

Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects

Oscar Serradilla[1] · Ekhi Zugasti[1] · Jon Rodriguez[2] · Urko Zurutuza[1]

Accepted: 12 November 2021 / Published online: 18 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Given the growing amount of industrial data in the 4th industrial revolution, deep learning solutions have become popular for predictive maintenance (PdM) tasks, which involve monitoring assets to anticipate their requirements and optimise maintenance tasks. However, given the large variety of such tasks in the literature, choosing the most suitable architecture for each use case is difficult. This work aims to facilitate this task by reviewing various state-of-the-art deep learning (DL) architectures and analysing how well they integrate with predictive maintenance stages to meet industrial companies' requirements from a PdM perspective. This review includes a self-organising map (SOM), one-class neural network (OC-NN) and generative techniques. This article explains how to adapt DL architectures to facilitate data variability handling, model adaptability and ensemble learning, all of which are characteristics relevant to industrial requirements. In addition, this review compares the results of state-of-the-art DL architectures on a publicly available dataset to facilitate reproducibility and replicability, enabling comparisons. Furthermore, this work covers the mitigation step with deep learning models, the final PdM stage that is essential for implementing PdM systems. Moreover, state-of-the-art deep learning architectures are categorised, analysed and compared; their industrial applications are presented; and an explanation of how to combine different architectures in a solution is presented that addresses their gaps. Finally, open challenges and possible future research paths are presented and supported in this review, and current research trends are identified.

Keywords Deep learning · Predictive maintenance · Data-driven · Survey · Review · Industry 4.0

Acronyms

AE	Autoencoder
AD	Anomaly detection
CM	Condition monitoring
CNN	Convolutional neural nework
DAE	Denoising autoencoder
DBN	Deep belief network
DL	Deep learning
ELM	Extreme learning machine
EMA	Exponential moving average
EOC	Environmental and operational conditions
FE	Feature engineering
FFNN	Feed forward neural network
GAN	Generative adversarial network
GRU	Gated recurrent unit
HI	Health index
LSTM	Long-short term memory
ML	Machine learning
MSE	Mean square error
NN	Neural network
OCC	One class classification
OC-SVM	One class support vector machine
PdM	Predictive maintenance
RBM	Restricted boltzmann machine
RCA	Root cause analysis
RMSE	Root mean square error
RNN	Recurrent neural network

✉ Oscar Serradilla
oserradilla@mondragon.edu

Ekhi Zugasti
ezugasti@mondragon.edu

Jon Rodriguez
j.rodriguez@koniker.coop

Urko Zurutuza
uzurutuza@mondragon.edu

1 Electronics and Computer Science, Mondragon Unibertsitatea, Loramendi 4, Mondragon, 20500, Spain

2 Koniker, San Andres auzoa 20, Mondragon, 20500, Spain

RUL	Remaining useful life
RVR	Relevance vector regression
SAE	Sparse autoencoder
SotA	State-of-the-art
SOM	Self organising map
SVR	Support vector regressor
VAE	Variational autoenconder
XAI	Explainable artificial intelligence

1 Introduction

In recent years, industry attention on artificial intelligence and machine learning (ML) techniques has risen due to their capacity to create automatic models that handle the large amounts of data currently collected, which is growing exponentially. Research into machine learning has switched to more complex models such as ensemble methods and deep learning (DL) due to their higher accuracy when applied to larger datasets. These methods have evolved due to increases in computing power — primarily advances in GPUs — making deep learning currently one of the most researched topics. and the latter mainly due to the evolution of GPU-s, being deep learning one of the most researched topics nowadays. These models have achieved state-of-the-art results in fields such as intrusion detection systems, computer vision and language processing.

Maintenance is defined by the norm EN 13306 [125] as *the combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function*. Moreover, EN 13306 defines three types of maintenance: improvement maintenance improves machine reliability, maintainability and safety while keeping the original function; preventive maintenance is performed before failures occur either in periodical or predictive ways; and corrective maintenance replaces defective/broken parts when a machine stops working. Currently, most industrial companies rely on periodical and corrective maintenance strategies.

However, industry is transitioning towards a fourth revolution, termed “Industry 4.0”, which is based on cyber physical systems and the industrial Internet of Things. Industry 4.0 combines software, sensors and intelligent control units to improve industrial processes and fulfill their requirements [80]. These techniques enable automatised predictive maintenance by analysing massive amounts of process and related data based on condition monitoring (CM).

Predictive maintenance (PdM) is the best maintenance type given its potential to achieve an overall equipment effectiveness (OEE) [127] above 90% by anticipating maintenance requirements [26, 29], promising a return on investment of up to 1000% [59]. Maintenance optimisation is a priority for industrial companies given that effective maintenance can reduce maintenance costs by to 60% by correcting machine, system and personal failures [28]. Concretely, PdM maximises components’ working lives by taking advantage of their unexploited lifetime potential while reducing downtime and replacement costs by performing replacements before failures occur, thus preventing expensive breakdowns and production time losses caused by unexpected stops.

The numerous research works on PdM can be classified into three approaches [73]: physical models, data-driven models and hybrid models. The physical model methods capitalise on prior system knowledge to build a mathematical description of system degradation [14, 65, 68, 95, 126]. It is easy to understand the physical meaning of these systems, but they are difficult to implement for complex systems.

Data-driven methods predict a systems’ state by monitoring its condition with solutions learned from historical data [13, 97, 148]. These methods are composed of statistical calculations, reliability functions and artificial intelligence methods. They are suitable for complex systems because they do not need to understand how the systems work. However, it is more difficult to relate their output to physical meaning.

Hybrid approaches combine the aforementioned two approaches [73, 155]. Data-driven and deep learning methods have gained popularity in industry in recent years due to improvements in machine data collection, which have enabled the development of accurate PdM models in complex systems.

1.1 Research methodology

The *research methodology* of this survey on *deep learning model applications for predictive maintenance* is provided in this paragraph. It is intended to identify trends, analyse significant works and detect future research lines.

Given that the number of publications in the field has increased exponentially in recent years, as exposed in Fig. 1, this survey covers studies published between 2016 and 2021. To conduct the research, we gathered information from various electronic database-search engines, including Scopus, Engineering Village, Springer Link, Science Direct, IEEE-Xplore, ACM Digital Library and Google Scholar. These resources provided access to different types of works, including high-impact journals and conference papers.

Given the high number of publications in the field, authors delimited the research space by defining keywords and research queries. Specifically, the terms “deep learning” AND “predictive maintenance” were the primary descriptors, grouped by predictive maintenance stages:

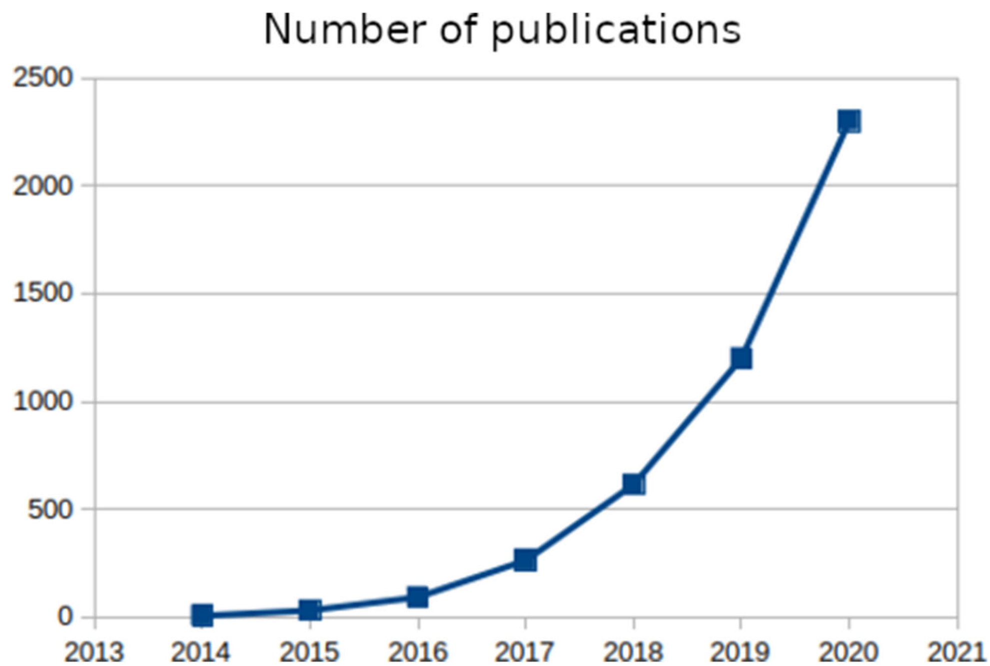

Fig. 1 Evolution of a number of publications on deep learning for predictive maintenance in Google Scholar search engine

"anomaly detection", "diagnosis", "prognosis", "mitigation" and their preparatory "preprocessing" and "feature engineering" stages, as presented in Fig. 2. In addition, complementary terms related to industrial requirements were also grouped with the primary descriptors (see Fig. 3): "transfer learning", "ensemble learning", "reinforcement learning" and "uncertainty modelling".

This work reviews 87 publications that address predictive maintenance stages using deep learning techniques, 19 works that combine deep learning and non deep learning data-driven algorithms to create architectures that better address PdM stages and 4 related review articles about deep learning applications for predictive maintenance: [31, 52, 153, 157].

1.2 Contributions

The goal of this survey is to provide an extensive review of deep learning techniques for predictive maintenance, specifying how these architectures can address each PdM stage by adapting to industrial requirements. Despite existing published reviews on machine learning and specifically deep learning for predictive maintenance e.g., [31, 52, 153, 157], this work provides the following *contributions* to the state-of-the-art (SotA):

(1) This work reviews state-of-the-art DL techniques for PdM, describes how they work, compares them and analyses them qualitatively. It also includes SOM, OC-NN and generative techniques, whose use in the PdM life-cycle has not previously been reviewed. (2) This work is oriented from a predictive maintenance problem perspective, focusing on how DL techniques implement each PdM stage to address industrial requirements. (3) This article explains how to adapt DL architectures to facilitate data variability handling, model adaptability and ensemble learning, and provides the relevant characteristics to address industrial requirements. (4) This work compares DL state-of-the-art results on a publicly available dataset to facilitate reproducibility and replicability, enabling comparisons. (5) This work covers the mitigation step with deep learning models, which is the final, essential PdM stage for PdM system implementation.

This paragraph describes the remaining content of this work. Section 2 reviews the background stages for predictive maintenance and provides an overview of the traditional data-driven models used in the field, together with an overview of deep learning techniques. Section 3 reviews and categorises the most relevant state-of-the-art deep learning works for predictive maintenance organised by underlying technique, analysing them by PdM stages to enable comparison. Moreover, related reviews are analysed and compared with this work to highlight the contributions of this work and how it addresses state-of-the-art gaps. Section 4 reviews the publicly available reference datasets for PdM model application and benchmarking. Section 5 discusses the suitability of deep learning models for predictive maintenance by evaluating their benefits and drawbacks and analysing the DL architectures qualitatively. Section 6 presents potential future research areas discovered during the elaboration of this research work. Finally, Section 7 concludes this survey by highlighting the most relevant aspects discovered during this work.

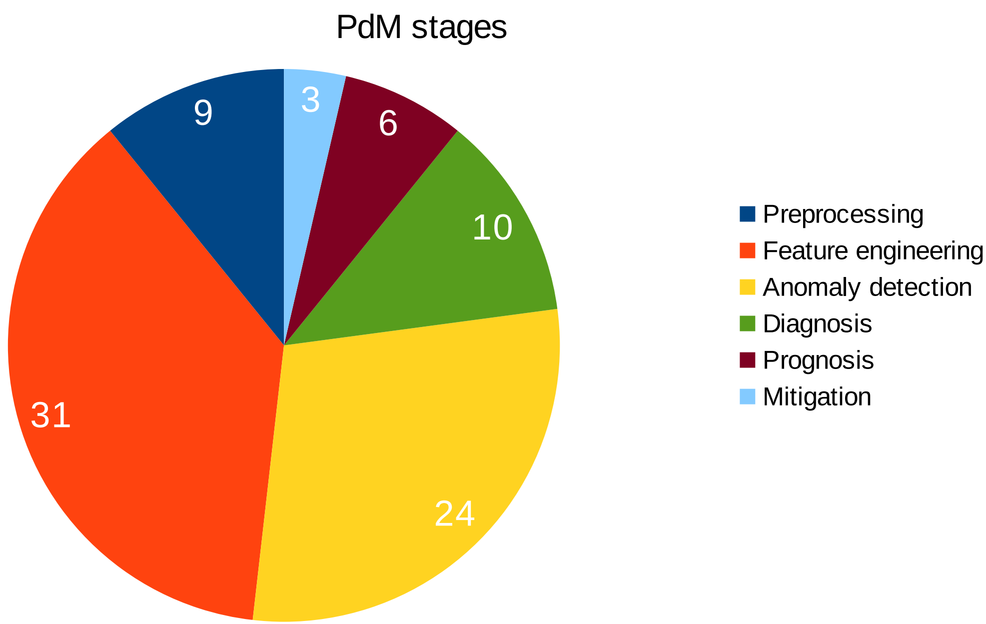

Fig. 2 The number of deep learning techniques articles by predictive maintenance stages

2 Overview of predictive maintenance and deep learning

2.1 Predictive maintenance background

Predictive maintenance solutions must consider many factors, peculiarities and challenges of industrial data, the most relevant of which are discussed in the subsequent paragraphs.

Venkatasubramanian et al. [126] presented 10 desirable properties for a PdM system: *quick detection and diagnosis, isolability (distinguishing among different failure types), robustness, novelty identifiability, classification error estimation, adaptability, explanation facility, minimal modelling requirements, real-time computation and storage handling, and multiple fault identifiability*.

Two main challenges of industrial use cases are their behaviour and data variability. These occur even in assets working under the same characteristics given the variations in mechanical tolerances, mount adjustments, variations in environmental and operational conditions (EOC) and other factors. These factors increase the difficulty of reusing PdM models among different machines and assets. Other relevant challenges are gathering quality data, performing correct preprocessing and feature engineering to obtain a representative dataset for the problem. In addition,

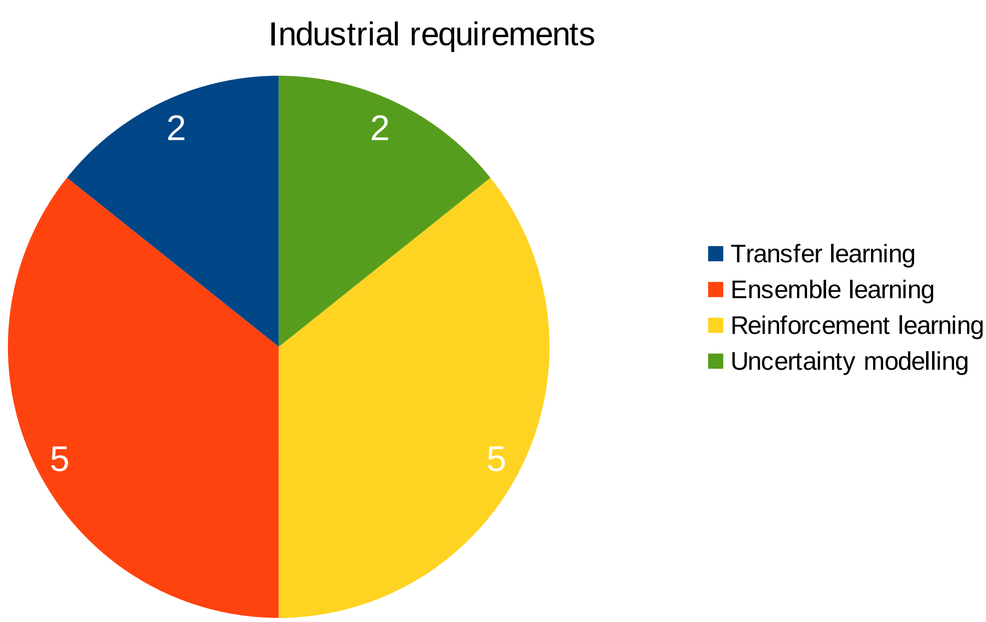

Fig. 3 A number of deep learning techniques that address industrial requirements by category

each observation is related to previous observations and therefore, they should be analysed together, which increases the data dimensionality and modelling complexity. Failure data gathering is difficult given that machines are designed and controlled to work correctly while preventing failures; therefore, such data are infrequent.

Some commonly monitored key components in PdM are (but are not limited to) bearings, blades, engines, valves, gears and cutting tools [153]. Some common failure types detected by CM are imbalance cracks, fatigue, abrasive and corrosion wear, rubbing, defects and leak detection, and others. The publication by Li and Gao [64] classifies the types of failures that may exist in the system as component failure, environmental impact, human mistakes and procedure handling.

The commonly used CM techniques are the following [123]: ultrasound [10], vibration analysis [92, 134], wear particle testing [12, 143], thermography, motor signal current analysis [30] and nondestructive testing [89], but additional techniques exist such as torque, voltage and envelopes [104], acoustic emission [49], pressure [156] and temperature monitoring [10, 156].

Environmental and operational conditions (EOCs) describe the working conditions for an industrial asset such a machine or component [122]. Environmental conditions refer to external conditions that affect these machines or components, such as ambient temperature or surrounding vibration perturbations. In contrast, operational conditions are working processes to which technical specifications are assigned, such as desired speeds, forces or positions. Additionally, machine data are monitored by sensors. When monitored and collected over time, these data comprise a dataset in the form of a time series. The analysis of such time series datasets using condition monitoring techniques enables the determination of component and machine states by comparing patterns and trends with historical data. The P-F curve [123] is a visual tool for presenting component degradation patterns in which health degrades from healthy working conditions to failure over time or as machine cycles progress.

2.2 Data-driven predictive maintenance stages

Deep learning models for PdM share the same principles as other machine learning and statistical techniques for PdM. Specifically, the data-driven methods that include deep learning for PdM follow the incremental steps presented in the roadmap shown in Fig. 4, which is based on the articles [102, 133] and the open system architecture for condition based maintenance standard OSA-CBM [60]: *anomaly detection*, *diagnosis*, *prognosis* and finally, *mitigation*.

To prepare the data for PdM, these methods perform two additional steps before the aforementioned ones, as presented in the general analytic lifecycle definition work [139], and PdM work [52]. These additional steps are preprocessing and feature engineering (FE), which, as stated above, are key enhancing model accuracy during the PdM stages by creating a representative dataset for the problem. All the PdM stages must be designed, adapted and implemented to fit specific use case requirements and data characteristics. In addition, PdM system development is incremental; therefore, the techniques, algorithms and decisions made during each stage will influence the following stages.

2.3 Deep learning techniques

This section presents the deep learning background and introduces the underlying structures that state-of-the-art PdM works use to create deep learning-based architectures. Information on how to create deep learning architectures for PdM and a review of publications in this field are presented in Section 3.

Currently, deep learning models outperform statistical and traditional ML models in many fields including PdM, when sufficient historical data exist. Deep learning architectures are based on neural networks that *go beyond shallow 1- and 2-hidden layer networks* [91].

Neural Networks (NNs) are formed by neurons that compute linear regressions of inputs with weights and then compute nonlinear activation functions such as sigmoid, rectified linear unit (ReLU) or tan-h to produce outputs. The network parameters are commonly initialised randomly, and are then adjusted to map the input data to the output data given the training dataset. This learning process occurs by running a gradient descending algorithm combined with a backpropagation algorithm. These enable calculations to adjust each neuron to reduce the error produced by the network; the error is calculated based on a user-defined cost function. The article by Kurt [46] justifies that NNs of at least two hidden layers with enough training data are capable of modelling any function or behaviour, creating the universal approximator.

The book by Goodfellow and Bengio [36] provides exhaustive background on DL and is considered a reference book in the field. Specifically, the book introduces machine learning and deep learning mathematical backgrounds. Afterwards, it focuses on DL optimisation, regularisation, different type of architectures, their mathematical definition and common applications. A simpler yet powerful overview of the field exists in the survey of DL applied to medicine by Litjens et al. [76], which is further complemented with a visual scheme that collects the main architectures. Another survey by Pouyanfar et al. [100] focuses specifically on DL architectures, applications, frameworks, SotA and historical works, trends and challenges. Additionally, the reference book on practical DL applications presented

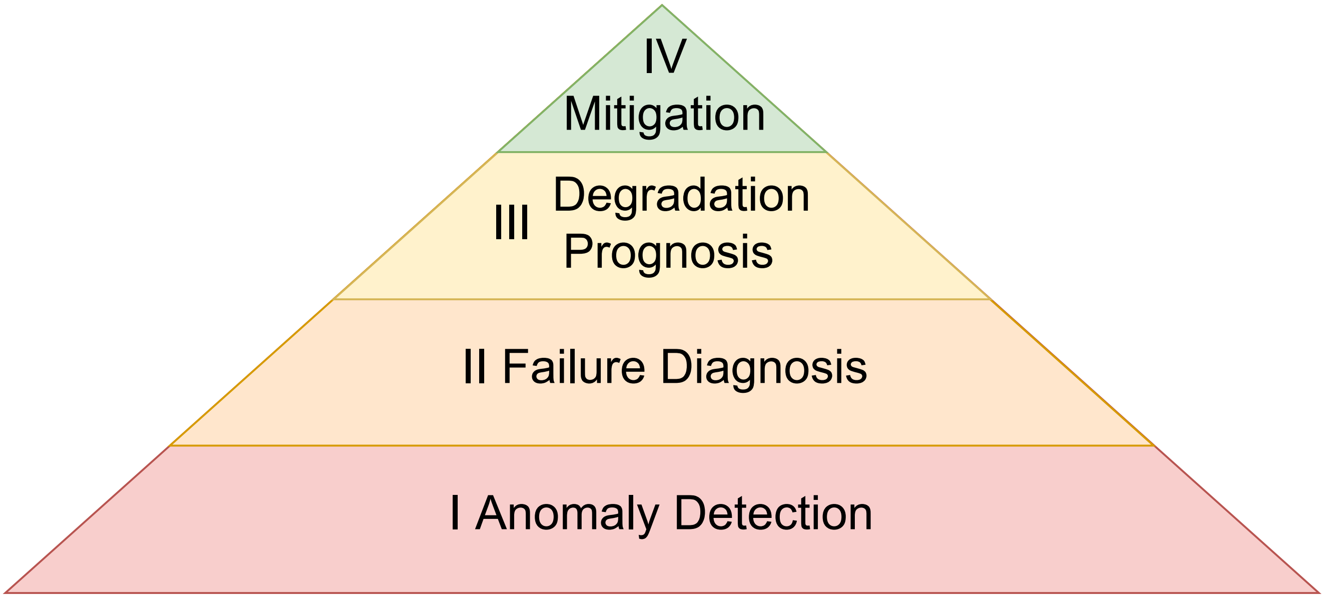

Fig. 4 Predictive maintenance roadmap stages

by Géron [33], is based on the Scikit-Learn, Keras, and TensorFlow tools.[1]

The most common DL *techniques* related to the field of PdM are summarised in the following paragraphs. Most are based on the feed-forward scheme, but each scheme has its own characteristics:

- The feed-forward neural network (FFNN) [137] is the first, most common and simplest architecture. It is formed by neurons stacked in layers, where the outputs of the neurons of one layer are connected to all the inputs of the neurons of the next layer. The neural network is provided with observations pairing input features and target features; the relations between these observations are learned by minimising the error produced by the network by mapping the input data to the output.
- A convolutional neural network (CNN) [61] is a type of feedforward network that maintains neurons' neighbourhoods by applying convolutional filters. CNNs have applications in image and signal recognition, recommendation systems and natural language processing, among others. The convolutional operation extracts features from the inputs and is usually fed to an FFNN for classification.
- A recurrent neural network (RNN) [108] models temporal data by saving the state derived from previous inputs of the network; however, RNNs often suffer from vanishing or exploding gradient problems [43], which cause these networks to forget long-term relations. To solve this problem, specific RNN architectures were created based on forget gates; these include long-short term memory (LSTM) [44] and gated recurrent unit (GRU) [25] models.
- The deep belief network (DBN) [42] and restricted boltzmann machine (RBM) [114] models are types of stochastic NNs that can learn a probability distribution over the data. They can be trained in either a supervised or unsupervised manner. Their main applications involve dimensionality reduction and classification.
- The autoencoder (AE) [11] is based on the singular value decomposition concept [35] to extract the nonlinear features that best represent the input data in a smaller space. AN AE consists of two parts: an encoder that maps input data to the encoded, and the decoder, which projects the latent space between them to a reconstructed space that has the same dimension as the input data. The network is trained to minimise the reconstruction error, which is the loss between the input and output. Different types of autoencoders exist that are employed for different use cases, as will be discussed later.
- Generative models such as the variational autoencoder (VAE) [53] and generative adversarial network (GAN) [37] were designed to work in an unsupervised way. A VAE is a generative and therefore nondeterministic modification of the vanilla AE in which the latent space is continuous. Usually, its latent space distribution is Gaussian, from which the decoder reconstructs the original signal based on random sampling and interpolation. A VAE has applications in estimating the data distribution, learning a representation of data

[1] Resources can be found in https://scikit-learn.org, https://keras.io and https://www.tensorflow.orgrespectively.

samples and generating synthetic samples, among others. A GAN is another type of generative neural network that consists of two parts: a generator and a discriminator. The generator is trained to generate an output that belongs to a specific data distribution using a representation vector as input. The discriminator is trained to classify whether its input data belongs to a specific data distribution. The generator's objective is to fool the discriminator by generating outputs from random input that cause the discriminator to classify it as belonging to the specific trained distribution.

– A Self-organising map (SOM) [55] is a neural network-based unsupervised way to organise the internal data representations. In contrast to typical neural networks that use backpropagation and gradient descent, a SOM uses competitive learning to create a new space called a map that is typically two-dimensional. It is based on neighbourhood functions that preserve the topological properties of the input space into the new space, represented in cells. It has applications in clustering, among others.

3 Deep learning for predictive maintenance

This section collects, summarises, classifies and compares the reference DL techniques for PdM by analysing the works and their applications. It includes accurate DL models that achieve SotA results from reviewed articles, surveys and reviews of the field. The works are classified by the principal DL technique used to perform each stage of Section 2.2 in the first six parts of this section. Additionally, more advanced DL architectures that combine different techniques or even perform more than one PdM stage simultaneously are reviewed in Section 3.7. Finally, the last subsection gathers the most relevant information contained in works similar to this survey by discussing the related reviews and surveys.

The reviewed works can be classified based on their underlying ML task and the algorithms used to address it, which are directly related to the use case and its data requirements. Binary classification is used when training data contain labelled failure and nonfailure observations. Multiclass classification is used in the same types of cases as binary classification, but where more than one failure type is classified; therefore, multiclass classification involves at least three classes: one represents nonfailure and then one class exists for each type of failure. One-class classification (OCC) is used when the training dataset contains only nonfailure data, which usually consists of machine data collected during early working states or when technicians ensure that the asset is working correctly. Finally, unsupervised techniques are used when the training datasets' observations are unlabelled; therefore, there is no knowledge of which observations belong to the failure or nonfailure classes. Unsupervised techniques can also be used as one-class classifiers. Additionally, there are a few works on other machine learning and deep learning topics, such as active learning, reinforcement learning and transfer learning.

3.1 Preprocessing

The initial step is to preprocess the data and prepare it for data-driven models by conducting techniques such as cleaning, encoding, imbalanced data handling and feature scaling, among others. Each PdM model has different requirements, and these must be taken into consideration when choosing adequate preprocessing techniques to boost model performance. Even though these techniques are not specific to the current field, common applications are explained to guide their use with deep learning-based PdM architectures. Complementary information on preprocessing techniques can be found in the article by Cernuda [18] on preprocessing for predictive maintenance.

Data cleaning is essential to obtain high-quality data. Its steps in predictive maintenance frequently imply handling missing values by imputation, such as interpolation or removing values, outlier handling, and ensuring that variables are in the expected range. This process can be enhanced by introducing domain expertise. In addition, neural networks have difficulties modelling categorical variables; therefore, these must be encoded into numerical values before they are input to the network; commonly, one neuron is created for each category.

Industrial companies have difficulties obtaining failure data; thus, they often lack sufficient failure data to train or test created models. This is why unsupervised and self-supervised architectures are becoming increasingly relevant in the predictive maintenance field. Nonetheless, after several failures have been collected, according to Mammado [83], two types of techniques exist that minimise this impact of this imbalanced data: data-level and algorithm-level techniques. The data-level methods are frequently oversampling methods; both SMOTE [20] and ADASYN [41] are widely used in predictive maintenance. Mammadov also states that algorithm-level methods adjust the classifier to fit imbalanced datasets, such as adjusting the misclassification costs or decision thresholds.

Two principal data scaling methods are used to prepare variables for deep learning models; these enable fair feature comparison and cause neural networks to be less sensitive to bias, according to the deep learning book by Géron [33] and a master's thesis on deep learning for PdM Silva [138]. One technique is min-max scaling (often termed as normalisation), which scales each variable to have values between the selected range by subtracting their minimum value and dividing the result by the maximum value minus the minimum value, as defined in (1). The other technique is standardisation, which transforms each variable to have a null expectation and unitary variance by subtracting their expectation and dividing by their variance. This technique is defined in (2).

$$X_i^S = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (1)$$

$$X_i^S = \frac{X_i - mean(X_i)}{var(X_i)} \quad (2)$$

The choice of preprocessing techniques is tied to data characteristics and conditioned by the selected deep learning architecture. One relevant factor for choosing the scaling technique is the activation functions used in the neural network; commonly, the scaling technique used is min-max scaling given that it ensures that the data are limited to the range expected by the network. Generally, when the network activation function is sigmoid, min-max scaling for the range [0, 1] is selected [131]; when tanh is used, the data are expected to be in the range $[-1, 1]$ [120], whereas when ReLU is used, the data are expected to be in the range $[0, inf]$, and therefore batch normalisation can be used in the previous layer [54]. Standardisation is less affected by outliers, but it does not bound values to specific ranges, as neural networks commonly require. In contrast, min-max scaling bounds the variables' range, although it is more sensitive to outliers.

3.2 Feature engineering

This step consists of extracting a relevant feature subset to be used as input for models in later stages. The deep learning algorithms used in PdM are capable of performing feature engineering automatically by obtaining a subset of the derived features best fit specifically for the task, which boosts model performance. A common technique is to use feed-forward methods by adding deep layers with fewer dimensions. RBMs also provide automatic feature extraction by modelling the data probability with contrastive divergence minimisation, which is based on one-way training and reconstructing the input from the output. Likewise, DBNs enable automatic feature extraction using stacked RBMs with greedy training, which can also be used for health index (HI) construction. Moreover, SOMs map data to a specified dimension, and AEs reduce dimensionality in latent space while preserving the maximum input data variance, providing nonlinear FE and HI calculations. In addition, CNNs automatically extract features by univariate or multivariate convolutions of the input, thus modelling sequential data with sliding windows. CNNs are usually combined with pooling methods to reduce dimensionality. Finally, RNNs use regression to model time-series and sequential data by propagating state information over time.

These feature engineering techniques remove the dependence on manual and feature engineering processes. Table 1 shows the strengths, limitations and referenced applications of the common deep learning techniques used for feature engineering. These techniques are integrated with machine learning and deep learning models to create architectures that can be applied to PdM stages.

Feed-forward networks are unable to model the temporal relations of industrial sensor data for feature extraction, but they can fuse nontemporal features to reduce the dimensionality of the feature set when used inside an AE. AEs have the ability to extract features automatically; therefore, they are suitable for extracting representative features to perform semisupervised and unsupervised predictive maintenance. However, like the feed-forward models and RBM, DBN and SOM, AEs depend on the use of CNN and RNN layers to extract time-based relations.

RBMs are simpler and faster to train than feed-forward networks, but they have difficulty in modelling complex industrial data because they are composed of a single layer. DBNs address this issue by stacking RBM layers; thus, they achieve SotA results in industrial data by modelling temporal relations with sliding windows. However, the use of sliding windows limits the long-term modelling capabilities of RBMs.

CNNs are suitable for modelling individual sensor relations with one-dimensional filters and can also model time-based relations among sensors by using two-dimensional filters. Their main advantage is that by weight sharing, they reduce the required training resources and model complexity, but they have limited memory. RNNs with specific architectures can extract longer temporal data relations among sensors, but their memory is still limited by the vanishing gradient problem. In addition, they add complexity and therefore increase the explanation difficulty of the network. Explanation difficulty is a challenge that PdM models must overcome before being deployed to production.

3.3 Anomaly detection

Anomaly detection aims to detect whether an asset is working correctly under normal conditions. Grouped by their underlying machine learning task, there are three ways to address this step using data-driven models: classification, one-class classification and clustering. These models can be used when labelled data for the different classes are available during the training phase, when only one class of data exists (commonly nonfailure data) and when the data are unlabelled, respectively.

Table 1 Deep learning techniques for automatic feature engineering and projection

Algorithm	Advantages	Disadvantages	Applications and references
Feed-forward models	- Reduce dimension to promote smaller feature space - Simplest NN architecture	- Do not model the features by neighbourhood - Do not model temporal relations	Engine health monitoring [103, 145], bearing fault diagnosis [2]
RBMs	- Preserve spatial representation in new space - Reduce training time	- Do not preserve data variance in the new space - Have difficulty modelling complex databecause they have only one layer	Bearing degradation [72], factory PLC sensors [47]
DBNs	- Competitive SotA results - Can model time-dependencies using sliding windows	- Lengthy training - Do not model long-term dependencies.	Vibration analysis [132], bearing prognosis [27], engines [98, 118], wind turbine [144]
SOMs	- Non-linear mapping of complex data to a lower dimension - Maintain feature distribution in the new space - Can be combined with other techniques for RCA (i.e., 5-whys [21])	- Have difficult linking latent variables with physical meaning - More complex than other techniques - Use a fixed number of clusters	Turbofan [58], pneumatic actuator [101], thermal power plant [21], bearing degradation [72]
AEs	- Automatic FE of raw sensor data achieveresults similar to traditional features[a] - Traditional features can also be input - No need for classification or failure data - Allows online CM.	- Extract features not specific to the task - Require more resources: both computational and training data - Lose temporal relations if input data are raw sensor data - Can lead to overfitting	Bearing vibration [1, 24, 45], satellite data [112], CAN vehicles [99]
CNN	- Simple yet effective - Faster than traditional ML models in production - Take advantage of neighbourhoods - Require less training time and data by weight-sharing - Can outperform LSTMs - Dropout can prevent overfitting	- Slower training due to the large number of weights - Analyse data in chunks and fail to model long-term dependencies.	Bearing diagnosis [17, 39], electric motor [77], gearbox [130], turbofan [9, 67], Numenta Anomaly Benchmark [87], blade [66]
RNNs	- Model temporal relationships of EOC data - Special architectures such as LSTM and GRU can model medium-term dependencies	- Can suffer from vanishing gradient problems; even special architectures cannot model very long-term depen dencies - Need more resources	turbofan [7, 16, 148], hydropower plant [147]

These techniques are based on input signal relations and temporal context

[a]In this work, the term traditional features refers to handcrafted and automatic feature extraction techniques such as statistical or ML-based features, excluding DL-based features

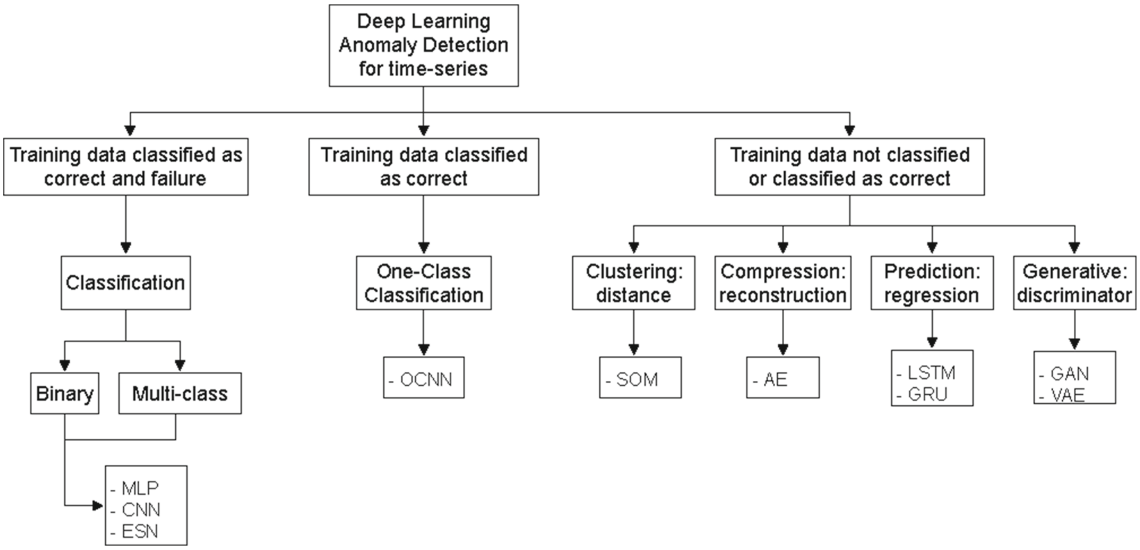

Fig. 5 The main deep learning techniques for anomaly detection in predictive maintenance

The deep learning-based AD algorithms can be classified into three groups based on the training data characteristics, as stated in the section introduction. The main architectures are summarised in Fig. 5.

These algorithms are summarised and compared, and their main applications are referenced in the subsequent tables. On the one hand, anomaly detection algorithms based on binary and multiclass classification approaches [2, 103] rely on training data classified as correct or failure. The commonly used feature extraction techniques are either traditional or deep learning features followed by a flattening process; then, several fully connected layers of decreasing dimension are applied until the output layer. For binary classification, one or two output neurons indicate the probability of failure or normal working conditions. Similarly, multiclass classifications have N+1 output neurons, where one neuron indicates the probability of not failing and each of the remaining N neurons indicates the probability of each type of failure.

On the other hand are the algorithms that address the AD problem based on one-class classification or unsupervised approaches using only training data classified as correct or unclassified. Autoencoder structures are widely used for this purpose, where vanilla AEs use a threshold in the reconstruction error and classify as anomalous data that surpasses that threshold. Stacking multiple AEs one after another is termed a stacked AE. SAEs constrain training with sparsity to keep neurons' activations low, and DAEs are AEs designed for noisy data. A generative VAE is an AE that maps input data to a posterior distribution, and GANs are used for data augmentation and AD in 2 ways: using a discriminator and using residuals.

One additional one-class technique is OC-NN, which trains an AE and freezes the encoder for one-class classification - similar to an OC-SVM loss function. Vanilla RNNs are also used for AD and analyse the tracking error between the predicted and actual behaviour using regression and measuring HI differences. Similarly, LSTM and GRU neural networks are used to replace the neuron architecture with LSTM and GRU neurons, respectively. A comparison of the strengths and limitations together with the applications and references of these techniques is shown in Table 2.

Autoencoders are trained to detect anomalies in industrial data using unsupervised or one-class data; a vanilla AE is the simplest version. Stacked AEs achieve better performances but at the cost of increased complexity and additional resources. SAEs penalise the weights of the autoencoder to limit complexity, which can be used to prevent overfitting of anomaly detection algorithms, and DAEs are more complex and robust to noisy data, making them suitable for addressing vibration data. An OC-NN works as a one-class neural network that can be trained in a semisupervised way. While it cannot extract time-based relations, this ability can be achieved by combining an OC-NN with CNN and RNN layers.

Regarding generative models, a VAE learns the posterior distribution of the sensor data, but the random component can make model interpretability difficult. GANs additionally enable data generation, which can be useful for generating synthetic failure data when only a few failure observations have been collected, and they can achieve SotA results in semisupervised anomaly detection. However, GANs have difficulties handling datasets with high imbalance ratios, their complexity makes them difficult for

Table 2 Anomaly detection methods that use training data classified as correct or unclassified: one-class and unsupervised classification

Algorithm	Advantages	Disadvantages	Applications and references
Autoencoders			
Vanilla AEs	- Automatic feature engineering of raw sensor data or traditional features - Minimise variance loss in latent space - No need for classification or failure data - Allows online CM	- Extract features not specific to the task - Require more resources, both computational and training data - Lose temporal relations if input data are raw sensor data - Can lead to overfitting	Bearing vibration [24, 45], flight data [106], CAN vehicles [99], marine autonomous systems [4]
Stacked AEs	- Perform slightly better than vanilla AEs	- Require more resources than vanilla AEs	Bearing vibration [110, 121], generator turbine vibration [32]
SAEs	- Same as AEs, but also prevent overfitting by forcing all neurons to learn	- Form more complex networks that require more resources than vanilla AEs	Bearing vibration, turbine vibration [1, 23, 32, 79]
DAEs	- Outperform vanilla AEs with noisy data - Work slightly better when several DAEs are stacked	- More complex networks that require more resources than vanilla AEs - Stacked DAEs need even more resources	Bearing vibration [79, 140]
Generative			
VAEs	- Learn posterior distribution from noisy distribution, generate data non-deterministically	- Implementation difficulties - Lose temporal relations when input data consist of raw sensor data.	Ball screw [134], electrostatic coalescer [82], web traffic [142], aircraft data [5]
GANs	- Good data augmentation with small imbalance ratio - ADs outperform unsupervised SotA methods	- Do not work well with large imbalance ratio - Complex and require more resources - May be outperformed by simpler methods such as CNN [17]	Induction motor [62], bearing multisensor [17]
One-Class Classifiers			
OC-NNs	- Automatic feature extraction	- Slower than traditional OCCs - Extracted features are not focused on the problem	General AD [19]
Recurrent Neural Networks			
Vanilla RNNs	- Model temporal relationships of time-series data - Self-learning.	- Suffer from vanishing gradient problems; therefore cannot model medium and long-term dependencies - Require more training resources than than do feedforward AEs or CNNs.	Activity recognition [6]
LSTMs	- Same as a vanilla RNN, however, these can model longer time dependencies than vanilla	- Even though these manage the vanishing gradient problem better than a vanilla RNN, they still have difficulty modelling long-term dependencies - Lengthy training and high computational requirements	Aircraft data [88], activity recognition [6], nuclear power mach-inery [146]
GRUs	- Comparable to LSTMs but easier to train	- Comparable to LSTMs	Aircraft data [88], activity recognition [6],

industrial stakeholders to interpret, and sometimes they are outperformed by simpler methods. RNNs are widely used to evaluate the evolution of industrial asset signals over time and detect anomalies, but the vanilla version cannot model long-term dependencies. LSTMs and GRUs fix this vanishing gradient problem, so they have currently replaced vanilla RNNs. The choice of one model type over the other depends on the specific use case being addressed.

3.4 Diagnosis

After an anomaly has been detected, the next stage involves diagnosing whether this anomaly belongs to a faulty working condition and will evolve into a future failure or whether, in contrast, there no failure risk exists. The diagnosis is usually based on root cause analysis (RCA) techniques, which aim to identify the true cause of a problem. The diagnosis algorithm must be suitable for the problem being addressed.

The diagnosis steps depend on the information and type of AD model used during the previous stage, given that PdM is an incremental process in which each stage is predicated on the previous stages. For multiclass classifiers, the type of failure related to the detected anomaly is already known; this characteristic enables a straightforward diagnosis and comparison with historical data [2, 103]. Nonetheless, most PdM architectures implement binary classifiers, one-class classifiers or unsupervised models, which lack failure-type information. Therefore, the results can be diagnosed only by grouping the detected anomalies by similarity, which is done using clustering models [3, 5, 8, 141, 159] and SOM [40, 69, 111, 115]. The features used during this stage are similar to those for AD; they can be based on either traditional or deep learning techniques.

3.5 Prognosis

After an anomaly has been detected and diagnosed, the degradation evolution can be monitored based on that moment's working conditions and machine state by focusing on the most influential features for the AD and diagnosis stages that can track failures. This step is usually carried out by remaining useful life (RUL) models that estimate the remaining time or cycles until a failure will occur when sufficient historical data for that failure type exists. Conversely, when the degradation data are insufficient, the only way to estimate the degradation is by tracking the evolution of HI or the distance between novel working states and the known good working states. Both aforementioned models can also provide a confidence bound.

The deep learning-based models for PdM prognosis are focused on fitting a regression model to prognosticate either the remaining useful life of the diagnosed failure or the health degradation when no historical data of that type exist. The RUL is commonly measured in time or by the number of cycles, while health degradation is tracked using anomaly deviation quantification by health indices. The most common algorithms are summarised and compared in Table 3. Vanilla RNNs and gate-based RNN networks (LSTM or GRU) can be used for regression, predicting features and HI evolution or predicting remaining cycles or time. Their inputs can be the information generated by previous stages as well as traditional or deep learning features. This section focuses on the most common and simple SotA techniques that use only DL for prognosis; prognosis works that combine DL with traditional features are presented in Section 3.7.

The use of LSTMs and GRUs is more common than that of vanilla RNNs given that they allow the modelling of longer time dependencies. LSTMs are more commonly used for prognosis in the PdM field, whereas GRUs achieve similar results but are simpler and therefore easier to train. The choice of one model type over the other depends on the addressed use case.

When target failure types are known and either a priori knowledge or observations of the target class exist and are available, uncertainty quantification can help in identifying which predictions of the generated model are trustworthy and which are not. This is particularly relevant for prognosis, because as the prediction time horizon increases, the prediction uncertainty rises. A common technique for quantifying the uncertainty of data-driven models is Bayesian inference, which is implemented in articles presented by Wang et al. [129] and Kraus and Feuerriegel [57]. However, when not enough data are collected from the target failure types or the task is approached as a one-class classification, the aforementioned techniques cannot be used. In this case, self-supervised metrics such as variance gain relevance for uncertainty modelling.

3.6 Mitigation

After an anomaly has been detected, its cause diagnosed and its remaining life prognosticated, there is enough information to perform maintenance actions to mitigate failures in early phases and thus prevent assets from degrading into failure. This stage consists of designing and performing the steps necessary to restore assets to correct working conditions before failures occur, which also reduces the implementation and downtime costs.

The research methodology followed in this publication showed few DL-based mitigation publications given that the majority of DL works focus on optimising a single performance metric, such as minimising error or maximising the anomaly detection rate, as stated in Sections 3.3,

Table 3 Summary of DL-based prognosis works for PdM

Algorithm	Advantages	Disadvantages	Applications and references
RNNs	Model temporal relationships of time-series data. Possibility for self-learning	Suffer from vanishing gradient problems; therefore, they cannot model medium and long-term dependencies. They have lengthy training and high computational requirements	Aero engine [148]
LSTMs	Same as a vanilla RNN; however, LSTMs can model longer time dependencies than can vanilla RNNs, and they outperform vanilla RNNs	Although LSTMs handle the vanishing gradient problem better than vanilla RNNs, they still have difficulty modelling long-term dependencies and have lengthy training and high computational requirements	Aero engine [148], rolling bearing [93, 149], lithium batteries [22, 154]
GRU	Same as LSTMs but easier to train	Same as LSTMs but may achieve slightly worse results	Aero engine, lithium batteries [22, 148]

The terms “unsup” and “sup” in the algorithm column refer to unsupervised and supervised respectively

3.4 and 3.5. Nonetheless, deep learning models are the most difficult ML type to understand given their higher complexity, which makes them more accurate at modelling high-dimensionality complex data; therefore, they fail to meet the industrial facility explanation requirement.

The publications that generate automatic data-driven maintenance policies using deep learning models for PdM are based on reinforcement learning, an emerging trend in this field. The article Paraschos et al. [96] uses reinforcement learning to generate control policies that optimise maintenance for degrading failure manufacturing systems. Moreover, Rocchetta et al. [109] presented a reinforcement learning framework to optimise power grid maintenance using Q-learning on a fully-connected neural network. Likewise, Ong et al. [94] proposed an automatic learning framework that creates optimal maintenance decision policies based on machine health state, derived from sensor data and proposes actionable recommendations.

Predictive maintenance systems should provide mitigation advice - or at least explanations - regarding the reasons why predictions were made, and such advice or explanations could be supported by the emerging field of explainable artificial intelligence (XAI). Furthermore, the final and most ambitious step in this PdM stage should be to automate recommendations for domain technicians to integrate PdM into the maintenance plan by optimising the industrial maintenance process via maintenance operation management.

3.7 Combination of models and remarkable works

The DL techniques already presented throughout the current section are the basic elements and architectures used for PdM. It is worth highlighting that infinite possible architectures are possible by combining these techniques or using them together with other data-driven or expert-knowledge-based techniques. The combination and adaptation of models for the problem being addressed results in more accurate models that fulfil its requirements.

This work reviews the principal deep learning works for PdM, even though the number of possible architectures is infinite by combining and adapting the presented techniques. Several common architectures of reviewed publications for anomaly detection, diagnosis and prognosis are presented in Figs. 6, 7 and 8, respectively.

The remainder of this subsection summarises the contributions and strengths of the relevant analysed works. One interesting article published by Shao et al. [117] presents a methodology of AE optimisation for rotating machinery fault diagnosis. First, they created a new loss function based on maximum correntropy to enhance feature learning. Second, they optimised the model’s key parameters to adapt it to signal features. This model was applied to fault diagnosis of gearbox and roller bearings. Another relevant publication by Lu et al. [78] uses growing SOM, an extension of the SOM algorithm that does not need specification of map dimension. This model was applied to simulated test cases with applications in PdM.

Guo et al. [38] proposed a model based on LSTM and an exponentially weighted moving average control chart for change point detection suitable for online training. An additional interesting work was presented by Lejon et al. [63], who used ML techniques to detect anomalies in hot stamping machines by non-ML experts. They aimed to detect anomalous strokes where the machine was not working properly. They presented the problem that most of the collected data correspond to press strokes of products without defects and that all the data are unlabelled. These data come from sensors that measure pressures, positions and temperature. The benchmarked algorithms were AE, OC-SVM and isolation forest, and AE outperformed the

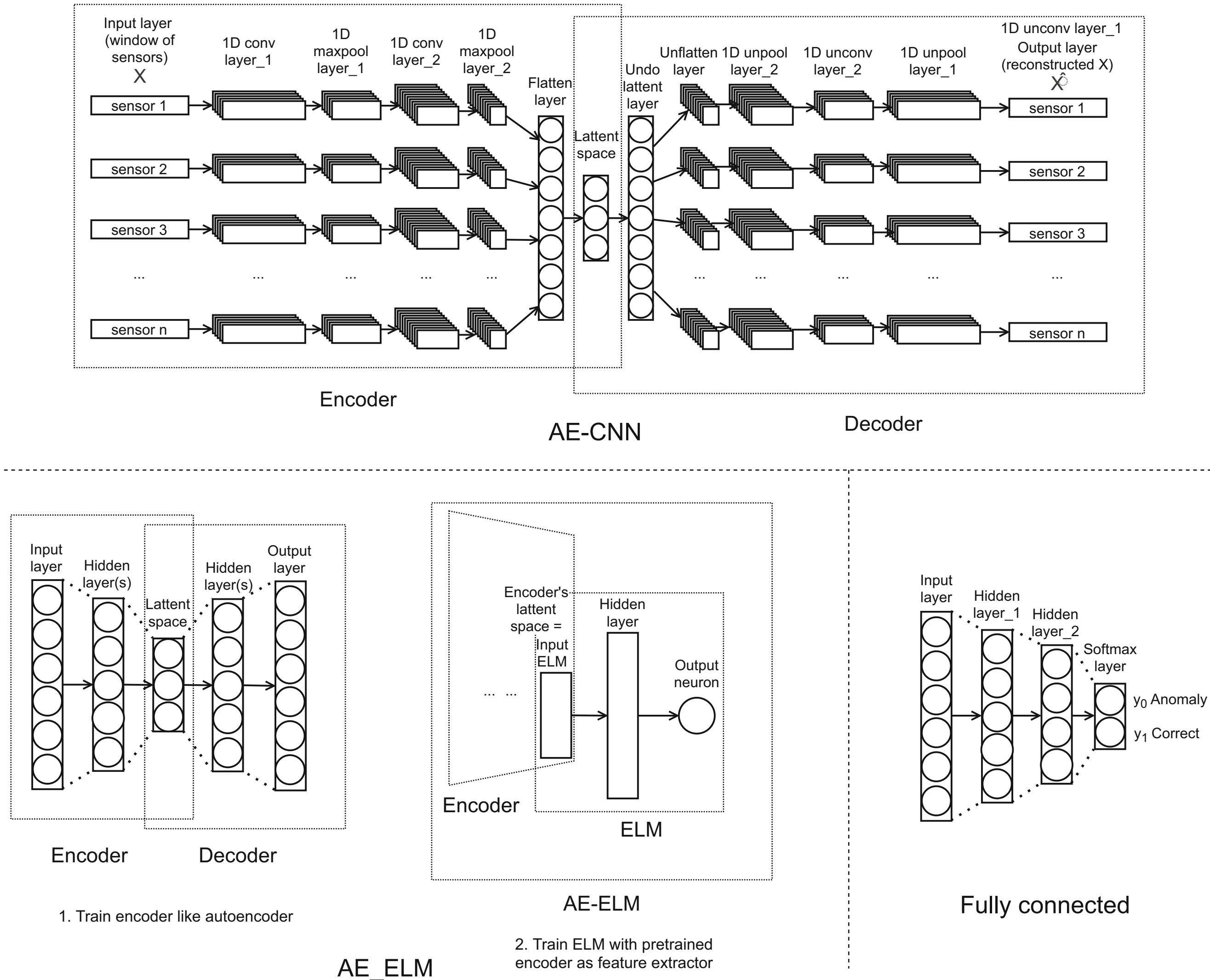

Fig. 6 Three common deep learning architectures for anomaly detection in predictive maintenance: convolutional autoencoder (top), autoencoder-based extreme learning machine (bottom left) and autoencoder-based ELM (bottom right)

rest, achieving the least number of false-positive instances. As the authors concluded, *the obtained results show the potential of ML in this field in transient and nonstationary signals when fault characteristics are unknown*, adding that AEs fulfill the requirements of low implementation cost and close to real-time operation, which will lead to more informed and effective decisions.

As previously mentioned in this article, the possibility of model combination is infinite. For instance the publication by Luo et al. [81], combines a GAN structure with LSTM neurons, two widely used DL techniques that achieve SotA results. Additionally, DL techniques can be combined with other computing techniques as discussed in Unal et al. [124], combining a feed forward network with genetic algorithms.

The last highlighted article that combines DL models is by Zhang et al.[152], and constitutes one of the most complete unsupervised PdM works. They built a model that uses the correlation of sensor signals in the form of signature matrices as input. This information is fed into an AE that uses a CNN and LSTM with attention for AD, partial RCA and RUL. The strengths of this work are the following: they show that correlation is a good descriptor for time-series signals, the attention mechanism using LSTMs provides temporal context, and the use of anomaly score as HI is useful for RCA, mapping the detected failures to the input sensors that originated them. However, their form of RCA is incomplete since they only correlate failures to input sensors but are not able to link them to physical meaning. Moreover, the lack of pooling layers together with the combination of DL techniques results in a complex and computationally expensive model that requires more time and data for training and yields decisions that are difficult to explain.

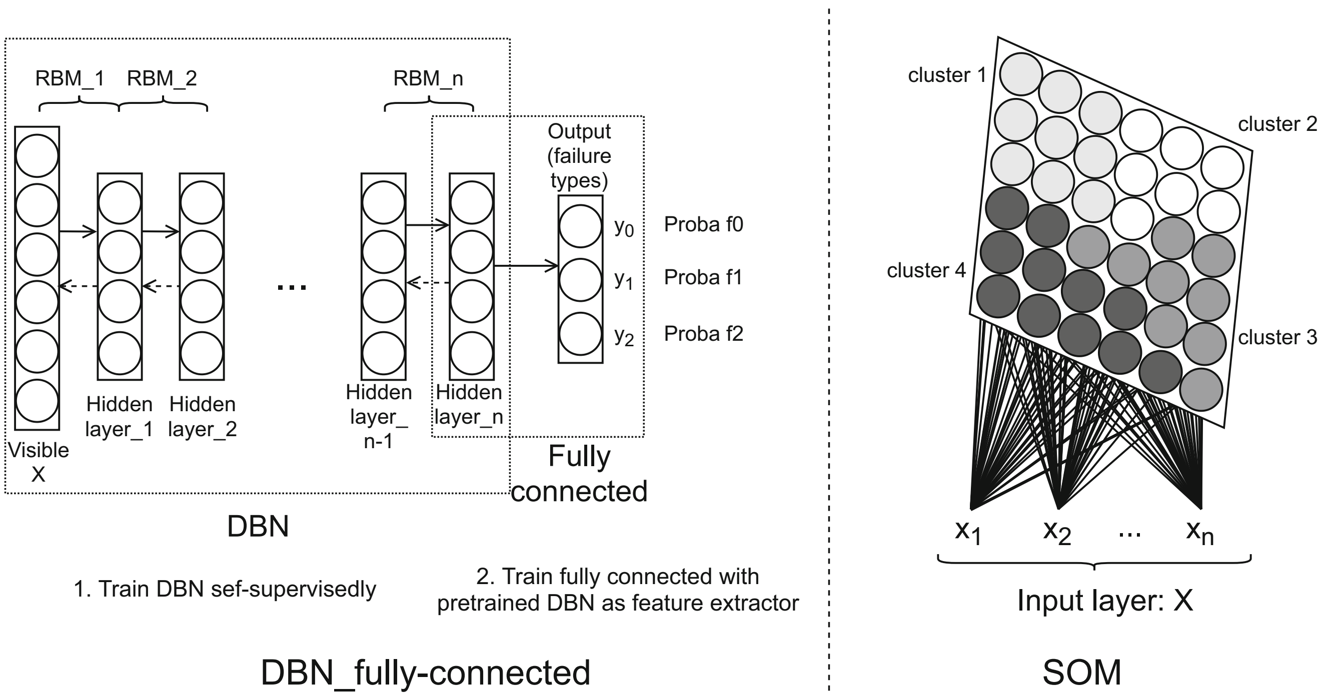

Fig. 7 Two common deep learning architectures for diagnosis in predictive maintenance: deep belief network with feed-forward predictor (left) and self organising map (right)

The following publications use other ML tasks combined with DL models for PdM, and other DL techniques. Wen et al. [135] used transfer learning with an SAE for motor vibration AD, outperforming DBNs. The article by Wen and Keyes [136] proposes a transfer learning based framework inspired by U-Net that is pretrained with univariate time-series synthetic data. The goal of this network is to be adaptable to other univariate or multivariate anomaly detection problems through fine-tuning.

Martínez-Arellano and Ratchev [85] presented a DL-based classifier using Bayesian search and CNN for AD. They first used a small labelled dataset to train the model

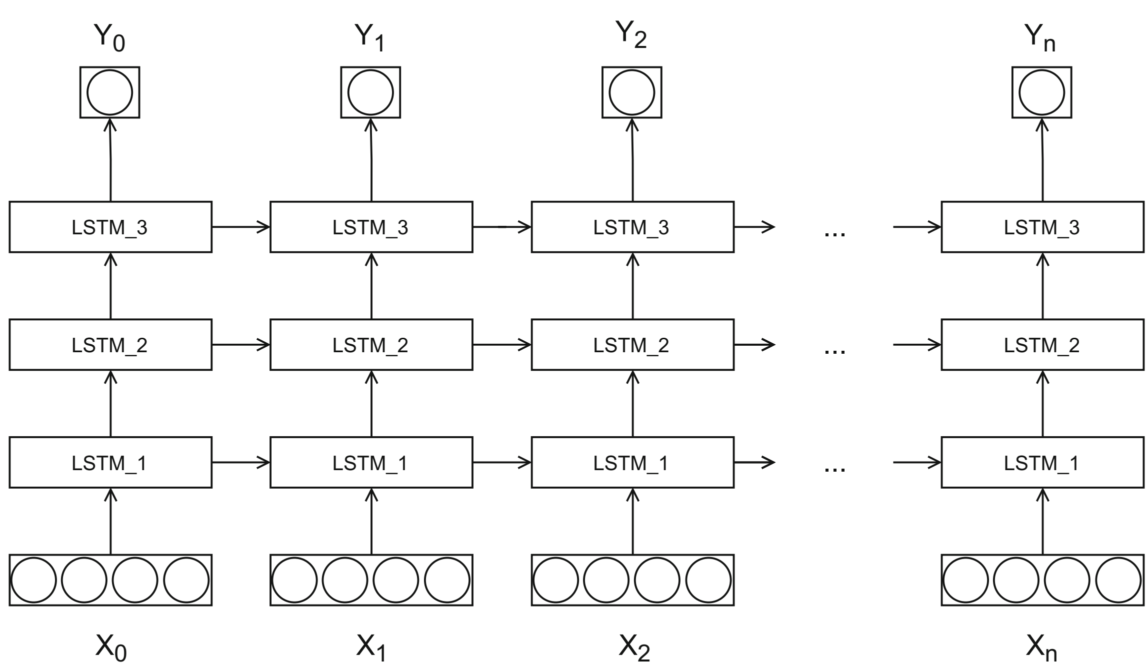

Fig. 8 A common deep learning architecture for predictive maintenance prognosis based on LSTM layers

and then used the model to classify the remaining data. The model uses uncertainty modelling to analyse the observations that cannot be correctly classified due to high entropy. Finally, it selects the top 100 with highest entropy to query a domain knowledge technician, asking him or her to label them to retrain the model with these new data. This procedure is followed until the model achieves good accuracy. This work is an example of how to use two interesting techniques in the field of PdM to address the problem of insufficient labelled data by querying domain technicians and showing them the instances from which the model can learn the most. Concretely, the aforementioned techniques belong to the semisupervised classification type using active learning. Similarly, the review by Khan and Yairi [52] mentions that expert knowledge can help troubleshoot the model and, if domain technicians are available, the model could learn from them using an ML training technique called active learning in which the model queries the technicians during the learning stage. Moreover, the work by Kateris et al. [51] uses SOM as the OCC model for AD together with active learning to progressively learn different fault stages.

The architectures of stacked autoencoders and stacked restricted Boltzmann machines mentioned above are commonly used to optimise the creation of more complex deep learning architectures by stacking one simple architecture type multiple times. However, little research has been applied to ensemble learning that combines different deep learning techniques for predictive maintenance or even with other data-driven systems. The article by Li et al. [70] trains the base algorithms separately and then uses a parallel ensemble method that weights the prediction of each base algorithm based on their performance to produce the output of the ensemble algorithm for aircraft data. The weight vectors are optimised using particle swarm optimisation and sequential quadratic optimisation algorithms. Similarly, the article by Li et al. [71] presents a method that weights the predictions of different remaining useful life algorithms and could be used to combine different deep learning models with themselves or other data-driven models. The work presented by Bose et al. [15] uses an ensemble-based voting system to create a one-class classifier relying on ELMs that optimises consumption and speeds up calculations; given the achieved neuron quantity reduction, this approach enables such models to be installed in edge computing scenarios.

Additionally, methods exist that fuse deep learning architectures, as proposed by Shao et al. [116], in which autoencoders are stacked based on majority voting, selective ensemble and weight assignment techniques for roller bearing diagnosis. Likewise, a stacked ensemble of recurrent neural networks to perform remaining useful life estimation was presented by Mashhadi et al. [86]. Overall, ensemble techniques have shown promising results in the field of predictive maintenance. However, the combination of algorithms in a meta-model increases the complexity and therefore makes explainability difficult, so the choice of whether to implementing ensemble methods is tied to the objectives of each use case.

Another interesting technique with PdM applications is deep reinforcement learning. Zhang et al. [150] uses it for HI learning, where it outperformed feed-forward networks but underperformed compared to CNN and LSTM for AD and RUL. This technique consists of transferring the knowledge acquired from one dataset to another dataset. The procedure consists of reusing a part or the complete pretrained model by adapting it to new requirements. While this approach sometimes requires retraining the model, it requires less data and time. In addition, Koprinkova-Hristova [56] used reinforcement learning on echo state networks to predict possible alarm situations in an industrial power plant, enabling model learning by experience, online readaptation from new information and human expert advice accounting.

3.8 Related review works summary

This subsection summarises the most relevant information of the review works related to this survey, highlighting their main contributions, detected challenges and gaps in the SotA works and their conclusions. In addition, Table 4 compares the contributions state-of-the-art reviews and surveys about deep learning-based PdM applications by analysing their applicability to PdM stages and their adaptability to relevant industrial requirements. Moreover, their description and limitations are presented and compared with the contributions of this article.

The work by Rieger et al. [107] conducts a qualitative narrative review on the SotA fast DL models applied for PdM in industrial IoT environments. They argue that real-time processing is essential for IoT applications, meaning that a high-latency system can lead to unintentional reactive maintenance due to insufficient maintenance planning time. Moreover, they highlight how DL models can be optimised. They state that weight sharing on RNNs enables parallel learning, which can help in training these types of networks that achieve SotA results in most PdM applications. Accordingly, they also justify the use of max-pooling layers when dealing with CNNs to eliminate redundant processing and thus optimise them.

Two DL reviews applied to other fields contain information about models that could be used for PdM: DL models for time series classification by Fawaz et al. [48] and DL used to model sensor data by Wang et al. [128]. However, these works do not focus on PdM, and therefore, their design, development and validation do not address predictive maintenance use case requirements.

The review by Zhao et al. [157] explains that there are algorithms that use traditional and handcrafted features, whereas others use DL features for the problem. It also

Table 4 Summary of related review works regarding DL application for PdM and comparisons with this article

	PdM stages					Industrial requirements							Description and limitations
	Compare PdM results	Anomaly detection	Diagnosis	Prognosis	Mitigation	Semi-supervised	Data variability	Adaptability	Transfer learning	Ensemble learning	Reinforcement learning	Uncertainty modelling	
Zhao et al. [157]	✓	✓	✓	✓	×	✓	✓	✓	×	×	×	×	Covers the main models: AE, RBM, DBN, CNN, RNN, but does not cover generative models. The results are compared quantitatively in a local dataset. Several techniques required to address industrial requirements are not covered.
Zhang et al. [153]	✓	✓	✓	✓	✓	✓	×	✓	×	×	✓	×	Only feed-forward and AE models are included. Their accuracy in different public datasets is presented. Several techniques required to address industrial requirements are not covered.
Khan and Yairi [52]	×	✓	✓	✓	×	✓	✓	✓	×	×	×	✓	Covers RBM, DBN, CNN, RNN, but does not cover generative models. It covers a few techniques required to address industrial requirements, but several are missing. It does not compare PdM results.
Fink et al. [31]	×	×	×	×	×	✓	×	×	✓	×	✓	✓	Reviews the main DL architectures including generative ones. It reviews the principal works, focusing on challenges. It does not compare architectures nor how they are applied to solve PdM stages. It includes a few techniques required to address industrial requirements, but several are missing.
This work	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Reviews the principal DL works by category, including one-class neural networks, SOM and generative models. Compares and discusses the results in a public dataset quantitatively. It also compares models qualitatively, which facilitates architecture fusion. Moreover, ensemble learning is reviewed to enable robust PdM models. The PdM mitigation step is presented, supported on domain technicians. It includes several techniques required to address industrial requirements that complement existing works.

The columns evaluate whether the works conduct a review of the corresponding characteristics presents the most common FE methods for DL-based PdM systems. The authors state that both aforementioned features work properly in DL models and are supported in their SotA revision. These works usually use techniques to boost model performance, such as data augmentation, model design and optimisation for the problem, and adopt architectures that already work in the SotA. They also adapt the learning function, apply regularizations, tweak the number of neurons and connections and apply transfer learning or stack models to enhance model generalisation and prevent overfitting. The advantage of traditional and handcrafted features is that they are not problem specific and are applicable to other problems. Moreover, they are easy for expert-knowledge technicians to understand given that they are based on mathematical equations. However, because they are not problem specific, in some cases, DL-based FE techniques perform better since these models are trained specifically for the problem and directly from the data. However, the results are not as intuitive as those using the aforementioned features, meaning that technicians can have difficulty understanding how they work.

The article by [157] also summarises the information already stated throughout this survey: DL models can achieve SotA results, pretraining in AEs can boost their performance, denoising models are beneficial for PdM because of the nature of sensor data, and CNN and LSTM variants can achieve SotA results in the field of PdM using model optimisation, depending on the scale of the dataset. In addition, domain knowledge can help in FE and model optimisation. Conversely, it is difficult to understand DL models despite various visualisation techniques because they are black-box models. Transfer learning could be used when few training data are available. PdM belongs to an imbalanced class problem because faulty data are scarce or missing.

The survey by [153] compares the accuracy obtained by several machine learning and deep learning architectures on different datasets and makes comparisons; however, because these comparisons are done with models applied to different datasets they are somewhat unfair. Nonetheless, they show high-accuracy results: most models reached accuracies of between 95% and 100%, emphasising that DL models can obtain promising results. They state that deeper models and higher dimensional feature vectors result in higher accuracy models, but require sufficient data. the increases in computational power and data growth in the field of PdM have tended to focus research on data-driven techniques, and specifically, DL models. However, the decisions of DL models lack explainability and interpretability.

The review by [52] states that the developed DL architectures are application or equipment-specific, and therefore, there is no clear way to select, design or implement those architectures. In addition, studies do not tend to justify the decision for selecting one architecture over another that also works for the problem, for instance, selecting a CNN versus an LSTM for RUL. The authors also argue that SotA algorithms, such as those presented throughout this section, have all been shown to work correctly and are similar. In addition, the work by [31] reviews relevant PdM works and current tendencies, but does not detail how to build DL-based models for each PdM stage.

Although this section focuses on DL models for PdM, we have seen that they are often integrated with traditional models and/or traditionally FE features, such as time and frequency domains, feature extraction based on expert knowledge or mathematical equations.

As the authors of [52] state, there is a lack of understanding of a problem when building DL models. They also argue that VAE is ideal for modelling complex systems because such models achieve high prediction accuracy without health status information. The algorithms that analyse the data while maintaining their time-series relationships by analysing the variables simultaneously are the most successful, regardless of whether a sliding window, a CNN or an LSTM technique is used. Most SotA algorithms focus on AD but can also be adapted to perform RUL by a regression or RNN, but the majority use LSTMs. Regressions commonly use features learned for the used AD models or even use traditional and handcrafted features. Generative models such as GAN do not work as well as expected. However, CNN works well while requiring less data and computing effort. This means that even DL models can achieve similar accuracy using traditional features or deep features extracted from the data in an unsupervised manner.

The majority of reviewed deep learning articles for PdM lack domain technician feedback, so they tackle the problem while relying only on data-driven techniques, without embracing domain knowledge. Moreover, few publications work on real industrial data given that industrial companies avoid publishing such data to protect them from competition. These facts comparing data-driven works difficult according to industrial requirements.

Overall, the existing reviews and surveys regarding DL applications for PdM have set the basis for current SotA works. However, this work complements the existing works and makes the following additional contributions: (1) This work takes a PdM problem perspective, focusing on how existing technologies address the PdM and industrial problems. In contrast, the related papers presented in Table 4 present the PdM concerns by focusing on technical perspectives. (2) This work explains all the state-of-the-art DL models for PdM and how they have been adapted to address the PdM stages, including the previously unreviewed models SOM and OC-NN, and it discusses model combination possibilities for creating architectures that better address use-case requirements. Moreover, it compares and discusses the differences among DL models qualitatively. In contrast, the existing publications review the main models, but omit several state-of-the-art models, such as generative models, whose use in the PdM life cycle has not been explained. (3) This work includes explanations for data variability handling, model adaptability or ensemble learning to ensure the robustness of deep learning models. These are relevant characteristics for PdM models that are not included in existing reviews. Most of the published reviews cover the semisupervised approach, include model adaptability to changes in EOCs and cover robustness to data variability, but only the work by Fink et al. [31] covers several recent topics, such as transfer learning and reinforcement learning. (4) This article complements the existing works by comparing state-of-the-art deep learning works for PdM on the widely researched public dataset turbofan [90], allowing replicability and comparison under the same criteria. In contrast, the existing works compare the results of different architectures in a local dataset, such as the work by Zhao et al. [157], or they present results of architectures in different datasets such as the work by Zhang et al. [153], which makes comparisons and replicability difficult. (5) This work includes a review of how DL-based PdM systems can be used to perform mitigation, a previously unreviewed PdM stage that is essential to ensure the success of PdM systems.

4 Comparison of state-of-the-art results

4.1 Benchmark datasets

The review by Khan and Yairi [52] states that one problem with PdM proposals is the lack of benchmarks, which makes comparisons difficult. Some public PdM datasets released by NASA are available for prognosis that were from the repository [90]. These datasets belong to the scope of predictive maintenance and are described in the following paragraphs.

The milling dataset [90] comprises acoustic emission, vibration and current sensor data acquired under different operating conditions and are intended for analysing the milling insert wear. Regarding the PdM stages, this dataset supports the application of AD, RCA and RUL.

The bearing dataset [90] consists of vibration data from 4 accelerometers monitoring bearings under constant pressure until failure. The result is a run-to-failure dataset in which all the failures occur after the design lifetime of 100 million revolutions has been exceeded. This dataset's possible PdM applications are AD and RUL estimation.

The turbofan engine degradation simulation dataset [90] contains run-to-failure data from engine sensors. Each instance starts at a random point in an engine's life at which it is working correctly and subsequently monitors its evolution until an anomaly occurs, after which the engine reaches a failure state. The engines are employed under different operational conditions and develop different failure modes. This dataset's possible PdM applications are AD, RCA and RUL.

The femto bearing dataset [90] is a bearing monitoring dataset from the Pronostia competition that contains run-to-failure and sudden failure data. The sensors used are thermocouples, which gathered temperature data, and accelerometers that monitored vibrations in the horizontal and vertical axes. The possible PdM applications of this dataset are AD, RCA and RUL.

To protect themselves from their competitors, industrial companies are reluctant to publish their own datasets because such datasets tend to reveal secret, private data and knowledge. A dataset that approximates data from most companies is published by the Semeion Research Center and named the steel plate fault dataset [74]; it contains steel plate faults classified into 7 categories.

4.2 Data-driven technique's results comparison

During the elaboration of this article, all the reviewed works aimed at anomaly detection and diagnosis used private datasets; therefore, they provide no opportunity to compare or replicate their results. However, the prognosis stage has been widely researched using the NASA turbofan dataset; thus, this stage has been used as a reference to compare model performance.

This subsection compares different relevant data-driven works for the PdM application turbofan dataset introduced in the previous subsection, which was generated using a *commercial modular aeropropulsion system simulation*. The reasons for choosing this dataset are that it is one of the reference datasets for PdM, enables the application of all PdM steps and is one of the most commonly used datasets for model ranking, although the majority of works focus on prognosis.

The challenge is divided into four datasets, each of which has different characteristics. The first, the FD001 dataset, contains 100 train and 100 test trajectories with one operational condition and a unique fault mode. The second, the FD002 dataset, contains 260 train and 259 test trajectories related to six operational conditions and unique fault modes. FD003 contains 100 train and 100 test trajectories with one operational condition and two different fault modes, and finally, FD004 contains 248 train and 249 test trajectories with six operational conditions and two different fault modes. All the datasets contain 3 operational setting variables and 26 sensors.

The dataset lacks an RUL label, which is the target column. Hence, this value is commonly assumed to be constant during the initial period of time when the system is working correctly and degrades linearly after exceeding the changepoint or initial anomalous point. The constant value during the initial period is a parameter denominated as R_{max}, which is set to values near 130 in many state-of-the-art works (see Table 5), enabling a fair comparison of their results.

The most common metrics for evaluating the models' performances are the following [9]: RMSE is the square root of the normalised sum of all the squared errors between real and predicted values, which penalises outliers more than does the mean absolute error. RMSE is defined in (3). The score function selected for this problem is suitable given that it is asymmetric and penalises later error predictions more than earlier ones. Concretely, it grows exponentially in distance from target value, but early predictions have lower exponent values than do later ones, which penalises the late predictions in (4), which was used in the PHM 2008 data challenge [113]. In the preceding equations, N is the number of engines in the test set, S is the computed score, and $h = (estimatedRUL - trueRUL)$.

Table 5 gathers the state-of-the-art results of data-driven models from 2014 on the four dataset subsets that use the presented two equations for model evaluation. As explained by Ramasso and Saxena [105], few works prior to 2014 used subset testing for model evaluation, and many used different performance metrics, which complicates comparisons. Therefore, we decided to omit those works and focus only on novel works that outperformed the results of previous works on at least one of the four data subsets.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} h_i^2} \tag{3}$$

$$S = \begin{cases} \sum_{i=1}^{N}\left(e^{-\frac{h_i}{13}} - 1\right) & for\ \ h_i < 0 \\ \sum_{i=1}^{N}\left(e^{\frac{h_i}{10}} - 1\right) & for\ \ h_i \geq 0 \end{cases} \tag{4}$$

The results comparison in Table 5 shows not only the models' performances but also the combination of preprocessing and feature engineering techniques. Therefore, the results show the performance of the complete data process applied to the dataset until prediction. Nonetheless, the table also shows that deep learning-based architectures have achieved state-of-the-art results in recent years. Concretely, these architectures are composed of combinations of different DL techniques.

Subset FD001 obtains lower errors; however, it contains only one operational condition and one failure type. Subset FD003 obtains similar results while containing two failure types. In contrast, the performances on subsets FD002 and FD004 are significantly higher given that operational conditions change each cycle during the same experiment. Therefore, it is normal for all algorithms to have significantly lower errors on subsets FD001 and FD003 compared with those on subsets FD002 and FD004.

5 Discussion

This section analyses deep learning architectures' applicability to the field of PdM. It contains a qualitative comparison of deep learning works on PdM, discusses the automatisation of their development and summarises their characteristics, advantages, drawbacks and main applications.. This section is the result of comparing the reviewed articles' trends, results and conclusions with PdM data characteristics and industrial requirements.

5.1 Qualitative comparison of deep learning in predictive maintenance

Different deep learning architecture types exist that can address PdM, as presented in Section 3. Each PdM use case has its own requirements; thus, the most suitable architecture for addressing these requirements should be selected based on their characteristics. Different deep learning techniques differ in complexity regarding their architecture type. Even models of the same type have differences in complexity due to their hyperparameters.

Autoencoders have the advantage of modelling sensor data in semisupervised and unsupervised scenarios, which are the most common PdM use cases. Their latent space representation can be used as new features for other data-driven models, and they are also applicable to anomaly detection by modelling the correct data class. Compared with other DL techniques, they have the advantage of simplicity, which reduces the required training resources and simplifies explanation tracking. Inside the autoencoder category, stacked autoencoders can facilitate training with respect to other autoencoders, helping to reduce the training loss of anomaly detection models for PdM. Sparse autoencoders can prevent the overfitting of anomaly detection PdM models, but they require a more complex network to perform the task than do vanilla autoencoders. Finally, a DAE has the capability to model noisy data and is commonly used to detect anomalies in vibration data to search for seizure and degradation failures.

Generative models are more complex than autoencoders, but they have advantages in semisupervised anomaly detection for PdM. VAEs infer the distribution of the training data, enabling the generation of synthetic samples of the original data distribution. They can handle noisy

Table 5 State-of-the-art results on four turbofan dataset subsets since 2014. The lower the metric, the better the model is considered to perform on average

Reference	R_{max}	Architecture	FD001 RMSE	FD002 RMSE	FD003 RMSE	FD004 RMSE	FD001 Score	FD002 Score	FD003 Score	FD004 Score
Ramasso and Saxena [105]	135	RUL-CLIPPER	13.3	22.9	16.0	24.3	*216*	2796	317	3132
Babu et al. [9]	130	FFNN	37.6	80.0	37.4	77.4	1.7e+4	7.8e+6	1.7e+4	5.6e+6
		SVR	21.0	42.0	21.0	45.3	1381	5.8e+5	1598	3.7e+5
		RVR	23.8	31.3	22.4	34.3	1504	1.7e+4	1431	2.6e+4
		DCNN	18.4	30.3	19.8	29.2	1287	1.3e+4	1596	7886
Zhang et al. [151]	130	MODBNE	15.0	25.1	12.5	28.7	334	5585	422	6558
Zheng et al. [158]	130	LSTM + FFNN	16.1	24.5	16.2	28.2	338	4450	852	5550
Li et al. [67]	125	CNN + FFNN	*12.6*	22.4	12.6	23.3	273	10412	284	1.2e+4
Listou Ellefsen et al. [75]	115-135	RBM + LSTM	*12.6*	22.7	*12.1*	22.7	231	3366	251	*2840*
Kakati et al. [50]	125	LSTM + attention	14.0	*17.7*	12.7	*20.2*	320	*2102*	*223*	3100

The best results are highlighted in bold sensor data better than can other autoencoders, but both architectures achieve similar results in other scenarios. In addition, VAE's stochastic approach increases the complexity of the model, and therefore, they are more difficult to explain than are vanilla autoencoders. Similarly, GANs can be used to create synthetic data from a learned distribution of the training data. They can also be used to detect anomalies in PdM through two techniques: by using the discriminator to detect observations not belonging to the machine's correct state or by setting a threshold on the residuals of the correct data and categorising observations that surpass this threshold as anomalous. Nonetheless, GANs form more complex models than do autoencoders; therefore, they are more difficult to explain.

CNNs are a good technique for modelling temporal relations in industrial data, but they must be combined with other stated architectures to perform feature extraction or anomaly detection. Their main drawback is that their memory is limited to the filter size. RNNs have also been widely used to model temporal relations in PdM. The most commonly used RNN architectures are LSTMs and GRUs, which achieve SotA results. Both obtain similar results in anomaly detection and prognosis; LSTMs achieve slightly better results, but GRUs contain a simpler structure and therefore have the advantage of faster training. These recurrent structures are also widely combined with autoencoders to facilitate modelling temporal relations, but this combination increases the complexity of autoencoders and makes them more difficult to explain.

The features extracted by any neural network can be used to reduce the input data dimensionality, which facilitates the use of SOMs, clustering techniques and XAI techniques for performing diagnosis on anomalies detected in a semisupervised way. The diagnosis of novel anomalies in PdM is particularly relevant given that these anomalies can be automatically detected by deep learning models, after which domain technicians can assist in their diagnosis. Technicians may then plan maintenance actions to restore the industrial asset's correct condition in the early anomalous stages, avoiding failure states.

5.2 Automatic development of deep learning models for predictive maintenance

Even though deep learning models can achieve SotA results in PdM datasets, their design, development and optimisation rely on related publications, author expertise and trial-and-error testing. Some of their biggest challenges are as follows: architecture type and structure choice, number of hidden layers and neurons, activation functions, regularisation terms to prevent overfitting and learning parameter optimisation.

For the above-stated reasons, the complete process of DL model creation is not as automatic as believed; this section aims to facilitate these tasks by explaining how state-of-the-art publications tackle them. To obtain competitive results, the authors preprocess and feature engineer the raw EOC signals. Such operations can boost model performance but simultaneously remove relevant information that could be learned automatically using more complex architectures. In addition, these steps are commonly performed by data scientists and do not embed domain knowledge; thus, the models are expected to learn all the nonlinear relations from the data. Conversely, this information could help in architectures' dimensionality reduction, resulting in simpler, more accurate and—as a result—more explainable models. Other by-product benefits are fewer training data requirements, less training time and higher generalisation to avoid overfitting.

One relevant factor when training deep learning models is the choice of loss function, which depends on the network architecture and data characteristics. The most common loss function used to train PdM neural networks is mean squared error (MSE), which is obtained by summing all the square differences between the predictions and their target values. This metric is mainly used for prognosis and unsupervised anomaly detection, given that minimising the MSE equals minimising the RMSE, which is a metric that averages errors by assigning more importance to outliers. The reason why MSE is more suitable than RMSE during training is that it removes the root square part of the equation, resulting in faster training. In the case of supervised neural networks for binary and multiclass classification, which is typical for supervised anomaly detection and diagnosis, the most common loss metric is cross-entropy. This metric is used similar to Kass in the article by Sleiman [119] to measure the differences among the probability distribution functions of the target classes: which are failure and not failure or even different failure types.

Different techniques exist in the literature to prevent overfitting and make networks generalise better. One typical method is to restrict network complexity to fit the training data, which reduces the number of layers and neurons, resulting in faster training and reducing overfitting issues. Another way to reduce the number of trainable parameters is to implement architectures that tie weights, such as CNNs or to use pooling layers (commonly max-pooling) to reduce dimensionality by obtaining only the most relevant information. Training different network parts in different steps with architectures such as stacked autoencoders and deep belief networks also facilitates training. Furthermore, regularisation techniques also reduce overfitting by conditioning weight evolution while preserving network structure. The dropout regularization randomly deactivates the output of each neuron at a specified probability for each training sample; thus, all neurons are forced to learn. Likewise, L1 and L2 regularisation terms can be added to the optimisation function to penalise large weights, given that these are related to overfitting.

The stated techniques can be combined with random initialization of small weights such as Xavier initialization by Glorot and Bengio [34] to prevent large weights, which increase the variance throughout the layers, causing vanishing gradients and preventing learning. Finally, adopting optimisation techniques such as learning rate decay and early termination help to halt network training at an optimal point before overfitting occurs.

The optimisation of deep learning architectures for PdM can be automatised by nonlinear optimisation algorithms, thus reducing the dependence on random and manual searches for architecture optimisation and hyperparameter tuning. The article by Martinez et al. [84] uses evolutionary optimisation by implementing genetic algorithms to optimise the architecture and parameters of a neural network for regression on rotorcraft vibrations. These parameters include the number of layers, number of neurons, number of filters in CNNs, or the number of LSTM networks. Similarly, the publication by Sleiman et al. [119] uses genetic algorithms for deep neural network optimisation to improve the accuracy of bearing diagnosis architecture.

5.3 Application of deep learning research in industrial processes

Most deep learning for predictive maintenance in the literature tackles PdM in an unsupervised way due to the difficulty of obtaining failure data from industrial companies. This is the reason that AEs, RBMs and generative models have so much repercussions in the field. The following paragraphs summarise the common techniques and how they meet industrial requirements.

Regarding SotA, a large number of DL proposals exist for AD and RUL. Most of these works tend to combine different algorithms to create more complex model that retains the advantages of the techniques that compose it. The most common combination for unsupervised PdM sensor modelling uses CNNs with LSTMs in an AE or AE-derived architecture. Similarly, supervised approaches usually use CNNs and LSTMs in a neural network that outputs the probability of failure types or regressions. However, such fusion techniques augment model complexity.

Regarding the diagnosis step, it is easy to perform RCA with supervised models given that—when the training data contain the label, failure, no failure, or even the type of failure—the model can directly map the new data to the corresponding failure type automatically. However, companies that lack this type of data can only model normality using OCC models or must even use an unsupervised approach to model unlabelled data. There is a gap in these latter models since they are unable to perform complete RCA given the impossibility of classifying unspecified failure types. One underlying reason could be the lack of collaboration between data scientists and expert-knowledge technicians. Therefore, this gap could be filled by applying explainable artificial intelligence techniques to facilitate the communication, understanding and guidance of DL models. XAI is a promising emergent field but has few publications in the field of PdM.

Deep learning models also fail to propose mitigation actions since, as mentioned before, they should work together with domain technicians' knowledge. However, the majority do not; they tackle the problem in a purely data-scientific way and ignore the underlying process working knowledge. For this reason, despite the accuracy of many models, they do not meet industrial and real PdM requirements. They present complex schemes with many hidden layers despite Venkatasubramanian et al. [126] stating that understandability is one desirable characteristic for PdM models. Without it, industrial companies may not deploy deep learning models to production, as domain technicians would be unable to understand their predictions and therefore, trust the models. Once again, the application of XAI techniques together with expert knowledge could overcome this problem by enabling technicians to understand the predictions, map detected failures to real physical root causes and even propose mitigation actions and give data-driven advice to help in maintenance management and with decision-making in manufacturing operation management.

The majority of the reviewed works were created and tested in research environments but not transferred to or tested in industrial companies. Although some models were trained with real industrial process data, the majority used reference datasets that were preprocessed and specifically prepared for the task, such as the ones presented in Section 4, which were generated in simulation or testing environments. However, the resulting models are unable to adapt to the requirements of industrial companies as presented by Venkatasubramanian et al. [126], which still prevail today. The work by Lejon et al. [63] consolidated the aforementioned needs by stating that industrial data are unlabelled and mostly correspond to non-anomalous process conditions. With regard to PdM architectures, the work by Khan and Yairi [52] seems to be the one that summarises and could better fit the requirements of the companies, even though it lacks any specification on how to address PdM in real companies.

Overall, we have seen that industrial companies need PdM models to be accurate, easy to understand, process streaming data and adapt to process data characteristics. Their data are mostly collected in an unsupervised way, or only nonfailure data are available. Moreover, such data are collected under different EOC. Consequently, there is a gap in the published data-driven models because the available unsupervised and OCC proposals are unable to link novel detected failures to their physical meaning, mainly because they ignore expert knowledge. In addition, few research publications exist on the application of XAI techniques in PdM, which could provide solutions for the main presented gaps.

As stated before, industrial companies that want to optimise their maintenance operations should transition towards predictive maintenance. However, this automatising should be embraced from simpler to more complex models, always choosing models that could better fit their specific needs. Both domain experts and data scientists should collaborate in the development and validation of a PdM structure. This mixture could benefit from the advantages of both domain knowledge-based and data-driven approaches, resulting in an accurate yet interpretable model. Explainable machine learning applied to deep learning could be an alternative to white- and grey-box models, which are more interpretable but less accurate. These new models may achieve a trade-off between accuracy and explainability by integrating with domain knowledge technicians, who can use them as tools for performing PdM and who can gain knowledge from the data while capitalizing on theoretical background and domain expertise.

6 Future research areas

The application of deep learning models for the development of predictive maintenance systems has grown in recent years. The reviewed works already cover techniques to address several industrial requirements. However, further research in several fields has the potential to develop advancements that address other industrial characteristics with DL models in PdM and improve how DL models address current maintenance requirements.

Given that industrial companies collect most of their data under normal working conditions, unsupervised and semisupervised methods are widely used to model the known data distributions and discover novel failure types. One research area that could facilitate addressing this imbalanced data problem is to simulate the modelled asset's behaviour. A realistic simulator could be obtained by the use of digital twins and could enable the simulation of different machine and component failure types.

Transfer learning is a research field that could simplify the life cycle of PdM systems and facilitate model reusability by reducing the required amount of data and training time to create PdM models, helping in adapting them to changes in EOCs. Moreover, transfer learning could help to reuse models learned with one component to components of the same type with similar characteristics.

The majority of deep learning works for PdM in recent years have focused on achieving highly accurate state-of-the-art results. However, other significant aspects of PdM requires further research, including interpretability, real-time execution and uncertainty modelling. Currently, there are emerging research trends that could address these mentioned gaps, such as combining explainable artificial intelligence and domain knowledge to interpret the behaviour of more accurate grey and black-box models; developing edge computing systems that integrate simplified architectures, reducing complexity to enable online data processing, and enriching model output with the probability for each prediction to model uncertainty. In addition, oversampling and data augmentation research on PdM will be useful when few faulty data are available, whereas in the meantime, generative models such as GANs and VAEs can fill this gap despite their higher complexity.

Another little researched area with promising potential is the diagnosis of semisupervised PdM systems given the necessity to perform RCA and classify novel failures by linking them to physical meaning. Changes to industrial working conditions require the adaptation of PdM systems, which can also be addressed by research on active learning and reinforcement learning techniques. Moreover, research on combining different data-driven techniques and ensemble learning will result in more robust models. Research on the aforementioned gaps and techniques is fundamental for transferring any machine learning model to real, industrial use cases and running them in production.

7 Conclusions

The majority of industrial companies that rely on corrective and periodical maintenance strategies can optimise maintenance costs by integrating automatic data-driven predictive maintenance models. These models monitor machine and component states, for which research has evolved from statistical to more complex machine learning techniques. Currently, the main research focus is on deep learning models.

The main objective of this survey was to analyse state-of-the-art deep learning technique implementation in the field of predictive maintenance. Consequently, several analyses and research are reviewed throughout the work. Initially, the most relevant factors and characteristics of industrial and PdM datasets were presented. Second, the steps necessary to perform PdM were presented methodologically. Then, various statistical and traditional machine learning techniques for PdM were reviewed to gain knowledge concerning the baseline models on which some deep learning implementations are based. Therefore, a thorough review of deep learning state-of-the-art works was conducted, the works were classified by their underlying techniques and data typology and then compared; which enabled the methods to be compared in a structured way. The related reviews on DL for PdM were also analysed, highlighting their main conclusions. Thereafter, a summary of the main public PdM datasets was presented, and the SotA results were compared on the turbofan engine degradation simulation dataset. Moreover, the suitability and the impacts of deep learning in the field of predictive maintenance were presented, together with a comparison with other data-driven methods. In addition, the systematisation of deep learning model development for predictive maintenance was discussed. Finally, the application of these models in real industrial use cases was argued, analysing their applicability beyond public benchmark datasets and research environments and highlighting the gaps between research architectures and industrial production requirements.

In summary, this survey presents a comprehensive review of deep learning techniques for predictive maintenance applications. Its main contributions to the state-of-the-art are as follows: a description of how DL techniques can solve each PdM stage in detail and an analysis of how to create DL architectures that can fit industrial requirements by applying currently researched techniques, such as transfer learning, reinforcement learning, uncertainty modelling and semisupervised approaches, while also addressing adaptability and data variability. In addition, a suitability analysis of DL for PdM and an analysis of their possible combination with other data-driven techniques is presented, including ensemble learning to create robust models. This article reviews the current publication trends, identifies their gaps and opens future lines of research.

Author Contributions This article has been written during the PhD thesis elaboration of the first author, who has done the principal research work; the second and last authors are the supervisors of the thesis, who have guided, supervised and reviewed the article; and the third author contributed with industrial expertise and review of the article.

Funding Oscar Serradilla, Ekhi Zugasti and Urko Zurutuza are part of the Intelligent Systems for Industrial Systems research group of Mondragon Unibertsitatea (IT1357-19), supported by the Department of Education, Universities and Research of the Basque Country. This work has been partially funded by the European Union's Horizon 2020 research and innovation programme's project QU4LITY under grant agreement n.825030, and by the Provincial Council of Gipuzkoa's project MEANER under grant agreement FA/OF 326/2020(ES).

Declarations

Competing interests This manuscript is the authors' original work and has not been published nor has it been submitted simultaneously elsewhere. All authors have checked the manuscript and have agreed to the submission, reporting no potential competing interests.

References

1. Ahmed HOA, Wong MLD, Nandi AK (2018) Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features. Mech Syst Sig Process 99:459–477. ISSN 10961216. https://doi.org/10.1016/j.ymssp.2017.06.027
2. Al-Raheem KF, Abdul-Karem W (2011) Rolling bearing fault diagnostics using artificial neural networks based on Laplace wavelet analysis. Int J Eng Sci Technol 2(6). ISSN 2141-2820. https://doi.org/10.4314/ijest.v2i6.63730
3. Amarbayasgalan T, Jargalsaikhan B, Ryu KH (2018) Unsupervised novelty detection using deep autoencoders with density based clustering. Appl Sci (Switzerland) 8(9):1468. ISSN 20763417. https://doi.org/10.3390/app8091468
4. Anderlini E, Salavasidis G, Harris CA, Wu P, Lorenzo A, Phillips AB, Thomas G (2021) A remote anomaly detection system for slocum underwater gliders. Ocean Eng 236:109531. ISSN 0029-8018. https://doi.org/10.1016/j.oceaneng.2021.109531
5. Chao MA, Adey BT, Fink O (2021) Implicit supervision for fault detection and segmentation of emerging fault types with deep variational autoencoders. Neurocomputing 454:324–338. ISSN 0925-2312. https://doi.org/10.1016/j.neucom.2021.04.122
6. Arifoglu D, Bouchachia A (2017) Activity recognition and abnormal behaviour detection with recurrent neural networks. Procedia Comput Sci 110:86–93. ISSN 18770509. https://doi.org/10.1016/j.procs.2017.06.121
7. Aydin O, Guldamlasioglu S (2017) Using LSTM networks to predict engine condition on large scale data processing framework. In: 2017 4th international conference on electrical and electronics engineering, ICEEE 2017. IEEE, pp 281–285. ISBN 9781509067886. https://doi.org/10.1109/ICEEE2.2017.7935834
8. Aytekin C, Ni X, Cricri F, Aksu E (2018) Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations. In: Proceedings of the international joint conference on neural networks, volume 2018-July. IEEE, pp 1–6. ISBN 9781509060146. https://doi.org/10.1109/IJCNN.2018.8489068
9. Babu GS, Zhao P, Li XL (2016) Deep convolutional neural network based regression approach for estimation of remaining useful life. In: Navathe SB, Wu W, Shekhar S, Du X, Wang XS, Xiong H (eds) Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 9642. Springer International Publishing, Cham, pp 214–228. ISBN 9783319320243. https://doi.org/10.1007/978-3-319-32025-0_14
10. Bakar AHA, Illias HA, Othman MK, Mokhlis H (2013) Identification of failure root causes using condition based monitoring data on a 33 kV switchgear. Int J Electr Power Energy Syst 47(1):305–312. ISSN 01420615. https://doi.org/10.1016/j.ijepes.2012.11.007
11. Ballard DH (1987) Modular learning in neural networks. In: AAAI, pp 279–284
12. Baptista A, Silva FJG, Pinto G, Porteiro J, Míguez J, Alexandre R, Sousa VFC (2021) Influence of the ball surface texture in the dragging of abrasive particles on micro-abrasion wear tests. Wear 476:203730. ISSN 0043-1648. https://doi.org/10.1016/j.wear.2021.203730, https://www.sciencedirect.com/science/article/pii/S0043164821001198. 23rd International Conference on Wear of Materials
13. Baptista M, Sankararaman S, de Medeiros IP, Nascimento C, Prendinger H, Henriques EMP (2018) Forecasting fault events

for predictive maintenance using data-driven techniques and ARMA modeling. Comput Ind Eng 115:41–53. ISSN 03608352. https://doi.org/10.1016/j.cie.2017.10.033
14. Blancke O, Komljenovic D, Tahan A, Combette A, Amyot N, Lévesque M, Hudon C, Zerhouni N (2018) A predictive maintenance approach for complex equipment based on petri net failure mechanism propagation model. In: Proceedings of the european conference of the PHM society, vol 4
15. Bose SK, Kar B, Roy M, Gopalakrishnan PK, Basu A (2019) AdepoS: Anomaly detection based power saving for predictive maintenance using edge computing. In: Proceedings of the Asia and south pacific design automation conference, ASP-DAC. ACM, pp 597–602. ISBN 9781450360074. https://doi.org/10.1145/3287624.3287716
16. Bruneo D, Vita FD (2019) On the use of LSTM networks for predictive maintenance in smart industries. In: Proceedings - IEEE international conference on smart computing, SMARTCOMP 2019. IEEE, pp 241–248. ISBN 9781728116891. https://doi.org/10.1109/SMARTCOMP.2019.00059
17. Rodríguez JPC (2019) Generative adversarial network based model for multi-domain. Universidad de Chile
18. Cernuda C (2019) On the relevance of preprocessing in predictive maintenance for dynamic systems. Predictive maintenance in dynamic systems: advanced methods, decision support tools and real-world applications, pp 53–92. https://doi.org/10.1007/978-3-030-05645-2_3
19. Chalapathy R, Menon AK, Chawla S (2018) Anomaly detection using one-class neural networks. arXiv preprint
20. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic minority over-sampling technique. J Artif Intell Res 16:321–357. ISSN 10769757. https://doi.org/10.1613/jair.953
21. Chemweno P, Morag I, Sheikhalishahi M, Pintelon L, Muchiri P, Wakiru J (2016) Development of a novel methodology for root cause analysis and selection of maintenance strategy for a thermal power plant: A data exploration approach. Eng Fail Anal 66:19–34. ISSN 13506307. https://doi.org/10.1016/j.engfailanal.2016.04.001
22. Chen J, Chen T-L, Liu W-J, Cheng CC, Li M-G (2021) Combining empirical mode decomposition and deep recurrent neural networks for predictive maintenance of lithium-ion battery. Adv Eng Inform 50:101405. ISSN 1474-0346. https://doi.org/10.1016/j.aei.2021.101405
23. Chen R, Chen S, He M, He D, Tang B (2017a) Rolling bearing fault severity identification using deep sparse auto-encoder network with noise added sample expansion. Proc Inst Mech Eng Part O J Risk Reliab 231(6):666–679. ISSN 17480078. https://doi.org/10.1177/1748006X17726452
24. Chen Z, Deng S, Chen X, Li C, Sanchez RV, Qin H (2017b) Deep neural networks-based rolling bearing fault diagnosis. Microelectron Reliab 75:327–333. ISSN 00262714. https://doi.org/10.1016/j.microrel.2017.03.006
25. Cho K, van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder–decoder approaches. Association for Computational Linguistics, Doha, Qatar, pp 103–111. https://doi.org/10.3115/v1/W14-4012
26. Colemen C, Damodaran S, Chandramoulin M, Deuel E (2017) Making maintenance smarter. Deloitte University Press, Westlake, Texas, pp 1–21
27. Deutsch J, He D (2017) Using deep learning-based approach to predict remaining useful life of rotating components. IEEE Trans Syst Man Cybern Syst 48(1):11–20. ISSN 21682232. https://doi.org/10.1109/TSMC.2017.2697842
28. Dhillon BS (2002) Engineering maintenance: a modern approach. CRC Press, Boca Raton
29. Sanger D (2017) Reactive preventive & predictive maintenance – IVC technologies
30. Dos Santos T, Ferreira FJTE, Pires JM, Damasio C (2017) Stator winding short-circuit fault diagnosis in induction motors using random forest. In: 2017 IEEE international electric machines and drives conference, IEMDC 2017, pp 1–8. ISBN 9781509042814. https://doi.org/10.1109/IEMDC.2017.8002350
31. Fink O, Wang Q, Svensén M, Dersin P, Lee W-J, Ducoffe M (2020) Potential, challenges and future directions for deep learning in prognostics and health management applications. Eng Appl Artif Intell 92:103678. ISSN 0952-1976. https://doi.org/10.1016/j.engappai.2020.103678
32. Galloway GS, Catterson VM, Fay T, Robb A, Love C (2016) Diagnosis of tidal turbine vibration data through deep neural networks. In: Proceedings of the third european conference of the prognostics and health management society, vol 2016, pp 172–180
33. Géron A (2017) Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems. O'Reilly Media. ISBN 9781491962299
34. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. J Mach Learn Res 9:249–256. ISSN 15324435
35. Golub GH, Reinsch C (1970) Singular value decomposition and least squares solutions. In: Numerische mathematik, vol 14. Springer, pp 403–420. https://doi.org/10.1007/BF02163027
36. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge. http://www.deeplearningbook.org
37. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Process Syst 3:2672–2680. https://doi.org/10.3156/jsoft.29.5_177_2
38. Guo T, Xu Z, Yao X, Chen H, Aberer K, Funaya K (2016a) Robust online time series prediction with recurrent neural networks. In: Proceedings - 3rd IEEE international conference on data science and advanced analytics, DSAA 2016. IEEE, pp 816–825. ISBN 9781509052066. https://doi.org/10.1109/DSAA.2016.92
39. Guo X, Chen L, Shen C (2016b) Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. Meas J Int Meas Confederation 93:490–502. ISSN 02632241. https://doi.org/10.1016/j.measurement.2016.07.054
40. Hao L, Xin X, Xiaojing W, Jiayu G, Jiexi S (2017) Health assessment of rolling bearing based on self-organizing map and restricted boltzmann machine. J Mech Transm (6):5
41. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the international joint conference on neural networks, pp 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969
42. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507. ISSN 00368075. https://doi.org/10.1126/science.1127647
43. Hochreiter S (1991) Untersuchungen zu dynamischen neuronalen Netzen. Master's thesis, Institut für Informatik, Techn Univ, Munchen, 91(1):1–71. ISSN 18168957 18163459
44. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
45. Hong S, Yin J (2018) Remaining useful life prediction of bearing based on deep perceptron neural networks. ACM Int Conf Proc Ser 48:175–179. https://doi.org/10.1145/3289430.3289438
46. Kurt H (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4(2):251–257
47. Hwang S, Jeong J, Kang Y (2018) SVM-RBM based predictive maintenance scheme for IoT-enabled smart factory. In: 2018 13th international conference on digital information management,

ICDIM 2018. IEEE, pp 162–167. ISBN 9781538652435. https://doi.org/10.1109/ICDIM.2018.8847132
48. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller PA (2019) Deep learning for time series classification: a review. Data Min Knowl Disc 33(4):917–963. ISSN 1573756X. https://doi.org/10.1007/s10618-019-00619-1
49. Jones MR, Rogers TJ, Worden K, Cross EJ (2022) A bayesian methodology for localising acoustic emission sources in complex structures. Mech Syst Sig Process 163:108143. ISSN 0888-3270. https://doi.org/10.1016/j.ymssp.2021.108143, https://www.sciencedirect.com/science/article/pii/S0888327021005239
50. Kakati P, Dandotiya D, Pal B (2019) Remaining useful life predictions for turbofan engine degradation using online long short-term memory network. In: ASME 2019 gas turbine india conference, GTINDIA 2019, vol 2, p 34. https://doi.org/10.1115/GTINDIA2019-2368
51. Kateris D, Moshou D, Pantazi XE, Gravalos I, Sawalhi N, Loutridis S (2014) A machine learning approach for the condition monitoring of rotating machinery. J Mech Sci Technol 28(1):61–71. ISSN 1738494X. https://doi.org/10.1007/s12206-013-1102-y
52. Khan S, Yairi T (2018) A review on the application of deep learning in system health management. Mech Syst Sig Process 107:241–265. ISSN 0888-3270. https://doi.org/10.1016/j.ymssp.2017.11.024
53. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: 2nd international conference on learning representations, ICLR 2014 - conference track proceedings, vol 1
54. Klein P, Weingarz N, Bergmann R (2020) Enhancing siamese neural networks through expert knowledge for predictive maintenance. In: IoT streams for data-driven predictive maintenance and IoT, Edge, and mobile for embedded machine learning, volume 1325 of communications in computer and information science. Springer International Publishing. ISBN 978-3-030-66769-6. https://doi.org/10.1007/978-3-030-66770-2_6, Accepted for publication
55. Kohonen T (1990) The self-organizing map. Proc IEEE 78(9):1464–1480
56. Koprinkova-Hristova P (2014) Reinforcement learning for predictive maintenance of industrial plants. Inf Technol Control 11(1):21–28. ISSN 1312-2622. https://doi.org/10.2478/itc-2013-0004
57. Kraus M, Feuerriegel S (2019) Forecasting remaining useful life: Interpretable deep learning approach via variational Bayesian inferences. Decisi Support Syst 125. ISSN 01679236. https://doi.org/10.1016/j.dss.2019.113100
58. Lacaille J, Gouby A, Bense W, Rabenoro T, Abdel-Sayed M (2015) Turbofan engine monitoring with health state identification and remaining useful life anticipation. Int J Cond Monit 5(2):8–16. ISSN 20476426. https://doi.org/10.1784/204764215815848375
59. Lavi Y (2018) The rewards and challenges of predictive maintenance. InfoQ
60. Lebold M, Reichard K, Byington CS, Orsagh R (2002) OSA-CBM architecture development with emphasis on XML implementations. In: Maintenance and reliability conference (MARCON), number January, pp 6–8. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.4066&rep=rep1&type=pdf
61. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551. ISSN 0899-7667. https://doi.org/10.1162/neco.1989.1.4.541
62. Lee Yh, Jo J, Hwang J (2017) Application of deep neural network and generative adversarial network to industrial maintenance: A case study of induction motor fault detection. In: Proceedings - 2017 IEEE international conference on big data, Big Data 2017, volume 2018-January. IEEE, pp 3248–3253. ISBN 9781538627143. https://doi.org/10.1109/BigData.2017.8258307
63. Lejon E, Kyösti P, Lindström J (2018) Machine learning for detection of anomalies in press-hardening: Selection of efficient methods. Procedia CIRP 72:1079–1083. ISSN 22128271. https://doi.org/10.1016/j.procir.2018.03.221
64. Li D, Gao J (2010) Study and application of reliability-centered maintenance considering radical maintenance. J Loss Prev Process Ind 23(5):622–629. ISSN 09504230. https://doi.org/10.1016/j.jlp.2010.06.008
65. Li L, Liu M, Shen W, Cheng G (2017) An expert knowledge-based dynamic maintenance task assignment model using discrete stress-strength interference theory. Knowl-Based Syst 131:135–148. ISSN 09507051. https://doi.org/10.1016/j.knosys.2017.06.008
66. Li P, Jia X, Feng J, Zhu F, Miller M, Chen LY, Lee J (2020) A novel scalable method for machine degradation assessment using deep convolutional neural network. Meas J Int Meas Confederation 151:107106. ISSN 02632241. https://doi.org/10.1016/j.measurement.2019.107106
67. Li X, Ding Q, Sun JQ (2018a) Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 172:1–11. ISSN 09518320. https://doi.org/10.1016/j.ress.2017.11.021
68. Li Y, Kurfess TR, Liang SY (2000) Stochastic prognostics for rolling element bearings. Mech Syst Sig Process 14(5):747–762. ISSN 08883270. https://doi.org/10.1006/mssp.2000.1301
69. Li Z, Fang H, Huang M, Wei Y, Zhang L (2018b) Data-driven bearing fault identification using improved hidden Markov model and self-organizing map. Comput Ind Eng 116:37–46. ISSN 03608352. https://doi.org/10.1016/j.cie.2017.12.002
70. Li Z, Goebel K, Wu D (2019a) Degradation modeling and remaining useful life prediction of aircraft engines using ensemble learning. J Eng Gas Turbines Power 141(4). ISSN 15288919. https://doi.org/10.1115/1.4041674
71. Li Z, Wu D, Hu C, Terpenny J (2019b) An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction. Reliab Eng Syst Saf 184:110–122. ISSN 09518320. https://doi.org/10.1016/j.ress.2017.12.016
72. Liao L, Jin W, Pavel R (2016) Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment. IEEE Trans Ind Electron 63(11):7076–7083. ISSN 02780046. https://doi.org/10.1109/TIE.2016.2586442
73. Liao L, Köttig F (2016) A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction. Appl Soft Comput J 44:191–199. ISSN 15684946. https://doi.org/10.1016/j.asoc.2016.03.013
74. Lichman M (2013) UCI machine learning repository
75. Listou Ellefsen A, Bjørlykhaug E, Æsøy V, Ushakov S, Zhang H (2019) Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab Eng Syst Saf 183:240–251. ISSN 09518320. https://doi.org/10.1016/j.ress.2018.11.027
76. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JAWM, van Ginneken B, Sánchez CI (2017) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88. ISSN 13618423. https://doi.org/10.1016/j.media.2017.07.005
77. Liu R, Meng G, Yang B, Sun C, Chen X (2017) Dislocated time series convolutional neural architecture: an intelligent fault diagnosis approach for electric machine. IEEE Trans Ind Inf 13(3):1310–1320. ISSN 15513203. https://doi.org/10.1109/TII.2016.2645238

78. Lu B, Stuber J, Edgar TF (2018) Data-driven adaptive multiple model system utilizing growing self-organizing maps. J Process Control 67:56–68. ISSN 09591524. https://doi.org/10.1016/j.jprocont.2017.06.006
79. Lu C, Wang ZY, Qin WL, Ma J (2017) Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. Sig Process 130:377–388. ISSN 01651684. https://doi.org/10.1016/j.sigpro.2016.07.028
80. Lukac D (2016) The fourth ICT-based industrial revolution "industry 4.0" - HMI and the case of CAE/CAD innovation with EPLAN P8. In: 2015 23rd telecommunications forum, TELFOR 2015. IEEE, pp 835–838. ISBN 9781509000548. https://doi.org/10.1109/TELFOR.2015.7377595
81. Luo Y, Cai Xi, ZHANG Y, Xu J, Yuanm X (2018) Multivariate time series imputation with generative adversarial networks. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds) Advances in neural information processing systems 31. Curran Associates, Inc., pp 1596–1607
82. Lygren S, Piantanida M, Amendola A (2019) Unsupervised, deep learning-based detection of failures in industrial equipments: The future of predictive maintenance. In: Society of petroleum engineers - Abu Dhabi international petroleum exhibition and conference 2019, ADIP 2019. Society of Petroleum Engineers. ISBN 9781613996720. https://doi.org/10.2118/197629-ms
83. Mammado EE (2019) Predictive maintenance of wind generators based on ai techniques. Master's thesis, University of Waterloo
84. Martinez D, Brewer W, Strelzoff A, Wilson A, Behm G, Wade D (2018) Deep learning evolutionary optimization for regression of rotorcraft vibrational spectra. In: Submitted to the 2018 symposium on machine learning for high performance computing environments MLHPC18
85. Martínez-Arellano G, Ratchev S (2019) Towards an active learning approach to tool condition monitoring with bayesian deep learning. Proc Eur Counc Model Simul, ECMS 33(1):223–229. ISSN 25222414. https://doi.org/10.7148/2019-0223
86. Mashhadi PS, Nowaczyk S, Pashami S (2020) Stacked ensemble of recurrent neural networks for predicting turbocharger remaining useful life. Appl Sci (Switzerland) 10(1). ISSN 20763417. https://doi.org/10.3390/app10010069
87. Munir M, Siddiqui SA, Dengel A, Ahmed S (2019) DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. IEEE Access 7:1991–2005. ISSN 21693536. https://doi.org/10.1109/ACCESS.2018.2886457
88. Nanduri A, Sherry L (2016) Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In: ICNS 2016: Securing an integrated CNS system to meet future challenges. IEEE, pp 5C2–1. ISBN 9781509021499. https://doi.org/10.1109/ICNSURV.2016.7486356
89. Narushin VG, Chausov MG, Shevchenko LV, Pylypenko AP, Davydovych VA, Romanov MN, Griffin DK (2021) Shell, a naturally engineered egg packaging: Estimated for strength by non-destructive testing for elastic deformation. Biosyst Eng 210:235–246. ISSN 1537-5110. https://doi.org/10.1016/j.biosystemseng.2021.08.023, https://www.sciencedirect.com/science/article/pii/S1537511021002075
90. NASA (2020) Prognostics center - data repository. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository
91. Neapolitan RE, Neapolitan RE (2018) Neural networks and deep learning. Determination Press, San Francisco. https://doi.org/10.1201/b22400-15
92. Nithyavathy N, Kumar SA, Sheriff KAI, Hariram A, Prasaad PH (2021) Vibration monitoring and analysis of ball bearing using gsd platform. Mater Today Proc 43:2290–2295. ISSN 2214-7853. https://doi.org/10.1016/j.matpr.2020.12.1088, https://www.sciencedirect.com/science/article/pii/S2214785320408065. International Conference on Advanced Materials Behavior and Characterization (ICAMBC 2020)
93. Niu Q (2017) Remaining useful life prediction of bearings based on health index recurrent neural network. Bol Tecnico/Tech Bull 55(16):585–590. ISSN 0376723X
94. Ong KSH, Niyato D, Yuen C (2020) Predictive maintenance for edge-based sensor networks: A deep reinforcement learning approach. In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). IEEE, pp 1–6
95. Oppenheimer CH, Loparo KA (2002) Physically based diagnosis and prognosis of cracked rotor shafts. In: Component and systems diagnostics, prognostics, and health management II, vol 4733, pp 122–132. International Society for Optics and Photonics, https://doi.org/10.1117/12.475502
96. Paraschos PD, Koulinas GK, Koulouriotis DE (2020) Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. J Manuf Syst 56:470–483. ISSN 02786125. https://doi.org/10.1016/j.jmsy.2020.07.004
97. Park J, Kim S, Choi J-H, Lee SH (2021) Frequency energy shift method for bearing fault prognosis using microphone sensor, vol 147, p 107068. ISSN 0888-3270. https://doi.org/10.1016/j.ymssp.2020.107068, https://www.sciencedirect.com/science/article/pii/S0888327020304544
98. Peng K, Jiao R, Dong R, Pi Y (2019) A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter, vol 361, pp 19–28. ISSN 18728286. https://doi.org/10.1016/j.neucom.2019.07.075
99. Lorenzo P (2019) Predictive maintenance for off-road vehicles based on Hidden Markov Models and Autoencoders for trend Anomaly Detection. PhD thesis, Politecnico di Torino
100. Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP, Shyu ML, Chen SC, Iyengar SS (2018) A survey on deep learning: Algorithms: techniques, and applications. ACM Comput Surv 51(5):92. ISSN 15577341. https://doi.org/10.1145/3234150
101. Prabakaran K, Kaushik S, Mouleeshuwarapprabu R (2014) Radial basis neural networks based fault detection and isolation scheme for pneumatic actuator. J Eng Comput Appl Sci 3(9):50–55
102. Prajapati A, Bechtel J, Ganesan S (2012) Condition based maintenance: a survey. J Qual Maint Eng 18(4):384–400. ISSN 13552511. https://doi.org/10.1108/13552511211281552
103. Rad MK, Torabizadeh M, Noshadi A (2011) Artificial neural network-based fault diagnostics of an electric motor using vibration monitoring. In: Proceedings 2011 international conference on transportation, mechanical, and electrical engineering, TMEE 2011. IEEE, pp 1512–1516. ISBN 9781457717017. https://doi.org/10.1109/TMEE.2011.6199495
104. Babu WR, Kumar RS, Kumar RS (2021) Rigorous investigation of stator current envelope of an induction motor using hilbert spectrum analysis. Mater Today Proc 45:2474–2478. ISSN 2214-7853. https://doi.org/10.1016/j.matpr.2020.11.054, https://www.sciencedirect.com/science/article/pii/S2214785320386557. International Conference on Advances in Materials Research - 2019
105. Ramasso E, Saxena A (2014) Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. Int J Prognostics Health Manag 5(2):1–15. ISSN 21532648
106. Reddy KK, Sarkar S, Venugopalan V, Giering M (2016) Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach. In: Proceedings of the annual conference of the prognostics and health management society, PHM, volume 2016-October, pp 192–199. ISBN 9781936263059

107. Rieger T, Regier S, Stengel I, Clarke N (2019) Fast predictive maintenance in Industrial Internet of Things (IIoT) with Deep Learning (DL) A review. In: CEUR workshop proceedings, vol 2348, pp 69–79
108. Robinson AJ, Fallside F (1987) The utility driven dynamic error propagation network. University of Cambridge Department of Engineering, Cambridge
109. Rocchetta R, Bellani L, Compare M, Zio E, Patelli E (2019) A reinforcement learning framework for optimal operation and maintenance of power grids. Appl Energy 241:291–301. ISSN 03062619. https://doi.org/10.1016/j.apenergy.2019.03.027
110. Roy M, Bose SK, Kar B, Gopalakrishnan PK, Basu A (2019) A stacked autoencoder neural network based automated feature extraction method for anomaly detection in on-line condition monitoring. In: Proceedings of the IEEE symposium series on computational intelligence, SSCI 2018. IEEE, pp 1501–1507. ISBN 9781538692769. https://doi.org/10.1109/SSCI.2018.8628810
111. Rustum R, Forrest S (2018) Fault detection in the activated sludge process using the kohonen self-organising map. In: 8th international conference on urban planning, architecture, civil and environment engineering, Dubai, UAE. https://doi.org/10.15242/heaig.h1217807
112. Sakurada M, Yairi T (2014) Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: ACM international conference proceeding series, volume 02-December-2014. ACM, pp 4–11. ISBN 9781450331593. https://doi.org/10.1145/2689746.2689747
113. Saxena A, Goebel K, Simon D, Eklund N (2008) Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 international conference on prognostics and health management, PHM 2008. https://doi.org/10.1109/PHM.2008.4711414
114. Scholar C, Smolensky P (1986) Parallel distributed processing: Explorations in the microstructure of cognition. Information Processing in Dynamical Systems: Foundations of Harmony Theory, vol 1. MIT Press, Cambridge, MA, USA, pp 194–281
115. Schwartz S, Jimenez JJM, Salaün M, Vingerhoeds R (2020) A fault mode identification methodology based on self-organizing map. Neural Computi Appl: 1–19. ISSN 14333058. https://doi.org/10.1007/s00521-019-04692-x
116. Shao H, Jiang H, Lin Y, Li X (2018) A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. Mech Syst Sig Process 102:278–297. ISSN 10961216. https://doi.org/10.1016/j.ymssp.2017.09.026
117. Shao H, Jiang H, Zhao H, Wang F (2017a) A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. Mech Syst Sig Process 95:187–204. ISSN 10961216. https://doi.org/10.1016/j.ymssp.2017.03.034
118. Shao SY, Sun WJ, Yan RQ, Wang P, Gao RX (2017b) A deep learning approach for fault diagnosis of induction motors in manufacturing. Chin J Mech Eng (Engl Ed) 30(6):1347–1356. ISSN 21928258. https://doi.org/10.1007/s10033-017-0189-y
119. Sleiman R, Raad A, Kass S, Antoni J, Antoni J Neuroevolution for bearing diagnosis
120. Soni H, Kansara A, Joshi T (2008) Predictive maintenance of gas turbine using prognosis approach
121. Tao S, Zhang T, Yang J, Wang X, Lu W (2015) Bearing fault diagnosis method based on stacked autoencoder and softmax regression. In: Chinese Control Conference, CCC, volume 2015-September. IEEE, pp 6331–6335. ISBN 9789881563897. https://doi.org/10.1109/ChiCC.2015.7260634
122. Tavner P, Li R, Penman J, Sedding H (2008) Condition monitoring of rotating electrical machines. Cond Monit Rotating Electr Mach: 1–250. https://doi.org/10.1049/PBPO056E
123. UESystems (2019) Understanding the P-F curve and its impact on reliability centered maintenance
124. Unal M, Onat M, Demetgul M, Kucuk H (2014) Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network. Meas J Int Meas Confederation 58:187–196. ISSN 02632241. https://doi.org/10.1016/j.measurement.2014.08.041
125. UNE-E N 13306 (2018) Maintenance. maintenance terminology. Standard, Asociación Española de Normalización, Génova, Madrid
126. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN (2003) A review of process fault detection and diagnosis part I: Quantitative model-based methods. Comput Chem Eng 27(3):293–311. ISSN 00981354. https://doi.org/10.1016/S0098-1354(02)00160-6
127. Vorne (2019) What Is OEE (Overall Equipment Effectiveness)
128. Wang J, Chen Y, Hao S, Peng X, Hu L (2019a) Deep learning for sensor-based activity recognition A survey. Pattern Recog Lett 119:3–11
129. Wang J, Liang Y, Zheng Y, Gao RX, Zhang F (2020) An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples. Renew Energy 145:642–650. ISSN 18790682. https://doi.org/10.1016/j.renene.2019.06.103
130. Wang PA, Yan R, Gao RX (2017a) Virtualization and deep recognition for system fault classification. J Manuf Syst 44:310–316. ISSN 02786125. https://doi.org/10.1016/j.jmsy.2017.04.012
131. Wang Q, Zheng S, Farahat A, Serita S, Gupta C (2019b) Remaining useful life estimation using functional data analysis. In: 2019 IEEE international conference on prognostics and health management, ICPHM 2019. https://doi.org/10.1109/ICPHM.2019.8819420
132. Wang X, Huang J, Ren G, Wang D (2017b) A hydraulic fault diagnosis method based on sliding-window spectrum feature and deep belief network. J Vibroengineering 19(6):4272–4284. ISSN 13928716. https://doi.org/10.21595/jve.2017.18549
133. Welz ZA (2017) Integrating disparate nuclear data sources for improved predictive maintenance modeling: Maintenance-based prognostics for long-term equipment operation, PhD thesis, University of Tennessee
134. Wen J, Gao H (2018) Degradation assessment for the ball screw with variational autoencoder and kernel density estimation. Adv Mech Eng 10(9). ISSN 16878140. https://doi.org/10.1177/1687814018797261
135. Wen L, Gao L, Li X (2019) A new deep transfer learning based on sparse auto-encoder for fault diagnosis. IEEE Trans Syst Man Cybern Syst 49(1):136–144. ISSN 21682232. https://doi.org/10.1109/TSMC.2017.2754287
136. Wen T, Keyes R (2019) Time series anomaly detection using convolutional neural networks and transfer learning. arXiv preprint
137. Werbos PJ (2005) Applications of advances in nonlinear sensitivity analysis. In: System modeling and optimization. Springer, pp 762–770. https://doi.org/10.1007/bfb0006203
138. Silva W (2019) CNN-PDM: A convolutional neural network framework for assets predictive maintenance
139. Wirth R, Hipp J (2000) Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, vol 1. Springer-Verlag, London, UK
140. Xia M, Li T, Liu L, Xu L, de Silva CW (2017) Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder. IET Sci, Meas Technol 11(6):687–695. ISSN 17518822. https://doi.org/10.1049/iet-smt.2016.0423

141. Xu F, Tse WtP, Tse YL (2018a) Roller bearing fault diagnosis using stacked denoising autoencoder in deep learning and Gath-Geva clustering algorithm without principal component analysis and data label. Appl Soft Comput J 73:898–913. ISSN 15684946. https://doi.org/10.1016/j.asoc.2018.09.037
142. Xu H, Feng Y, Chen J, Wang Z, Qiao H, Chen W, Zhao N, Li Z, Bu J, Li Z, Liu Y, Zhao Y, Pei D (2018b) Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in Web applications. In: Proceedings of the 2018 World Wide Web Conference, pp 187–196. https://doi.org/10.1145/3178876.3185996
143. Yadav PK, Dixit G, Dixit S, Singh VP, Patel SK, Purohit R, Kuriachen B (2021) Effect of eutectic silicon and silicon carbide particles on high stress scratching wear of aluminium composite for various testing parameters. Wear 482-483:203921. ISSN 0043-1648. https://doi.org/10.1016/j.wear.2021.203921, https://www.sciencedirect.com/science/article/pii/S0043164821003100
144. Yang W, Liu C, Jiang D (2018) An unsupervised spatiotemporal graphical modeling approach for wind turbine condition monitoring. Renew Energy 127:230–241. ISSN 18790682. https://doi.org/10.1016/j.renene.2018.04.059
145. Yildirim MT, Kurt B (2016) Engine health monitoring in an aircraft by using Levenberg-marquardt feedforward neural network and radial basis function network. In: Proceedings of the 2016 international symposium on INnovations in intelligent systems and applications, INISTA 2016. IEEE, pp 1–5. ISBN 9781467399104. https://doi.org/10.1109/INISTA.2016.7571847
146. You D, Shen X, Liu G, Wang G (2021) Signal anomaly identification strategy based on bayesian inference for nuclear power machinery. Mech Syst Sig Process 161:107967. ISSN 0888-3270. https://doi.org/10.1016/j.ymssp.2021.107967
147. Yuan J, Wang Y, Wang K (2019) LSTM based prediction and time-temperature varying rate fusion for hydropower plant anomaly detection: A case study. In: Lecture notes in electrical engineering, vol 484. Springer, pp 86–94. ISBN 9789811323744. https://doi.org/10.1007/978-981-13-2375-1_13
148. Yuan M, Wu Y, Lin L (2016) Fault diagnosis remaining useful life estimation of aero engine using LSTM neural network. In: AUS 2016 -2016 IEEE/ CSAA international conference on aircraft utility systems. IEEE, pp 135–140. ISBN 9781509010875. https://doi.org/10.1109/AUS.2016.7748035
149. Zhang B, Zhang S, Li W (2019a) Bearing performance degradation assessment using long short-term memory recurrent network. Comput Ind 106:14–29. ISSN 01663615. https://doi.org/10.1016/j.compind.2018.12.016
150. Zhang C, Gupta C, Farahat A, Ristovski K, Ghosh D (2019b) Equipment health indicator learning using deep reinforcement learning. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in Bioinformatics), LNAI, vol 11053. Springer, pp 488–504. ISBN 9783030109967. https://doi.org/10.1007/978-3-030-10997-4_30
151. Zhang C, Lim P, Qin AK, Tan KC (2017) Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Trans Neural Netw Learn Syst 28(10):2306–2318. ISSN 21622388. https://doi.org/10.1109/TNNLS.2016.2582798
152. Zhang C, Song D, Chen Y, Feng X, Lumezanu C, Cheng W, Ni J, Bo Z, Chen H, Chawla NV (2019c) A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 1409–1416. https://doi.org/10.1609/aaai.v33i01.33011409
153. Zhang W, Yang D, Wang H (2019d) Data-driven methods for predictive maintenance of industrial equipment: a survey. IEEE Syst J 13(3):2213–2227. ISSN 19379234. https://doi.org/10.1109/JSYST.2019.2905565
154. Zhang Y, Xiong R, He H, Pecht MG (2018) Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Trans Veh Technol 67(7):5695–5705. ISSN 00189545. https://doi.org/10.1109/TVT.2018.2805189
155. Zhao F, Tian Z, Zeng Y (2013) Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method. IEEE Trans Reliab 62(1):146–159. ISSN 00189529. https://doi.org/10.1109/TR.2013.2241216
156. Zhao P, Kurihara M, Tanaka J, Noda T, Chikuma S, Suzuki T (2017) Advanced correlation-based anomaly detection method for predictive maintenance. In: 2017 IEEE international conference on prognostics and health management, ICPHM 2017. IEEE, pp 78–83. ISBN 9781509057108. https://doi.org/10.1109/ICPHM.2017.7998309
157. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX (2019) Deep learning and its applications to machine health monitoring. Mech Syst Sig Process 115:213–237. ISSN 10961216. https://doi.org/10.1016/j.ymssp.2018.05.050
158. Zheng S, Ristovski K, Farahat A, Gupta C (2017) Long short-term memory network for remaining useful life estimation. In: 2017 IEEE international conference on prognostics and health management, ICPHM 2017. IEEE, pp 88–95. ISBN 9781509057108. https://doi.org/10.1109/ICPHM.2017.7998311
159. Bo Z, Qi S, Min MR, Cheng W, Lumezanu C, Cho D, Chen H (2018) Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In: 6th international conference on learning representations ICLR 2018 - Conference track proceedings

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Oscar Serradilla is a researcher and lecturer in the Research Group of Intelligent Systems for Industrial Systems of Mondragon Unibertsitea, where he teaches, designs, develops and implements projects related to data analysis, the Big Data paradigm and predictive maintenance. Oscar holds a degree in Computer Engineering from Mondragon Unibertsitatea and a Master in Computational Engineering and Intelligent Systems from the University of the Basque Country. Between 2014 and 2016 he was the creator of Pothole Avoider, a road safety application with which he participated 2 times in the Talentum Startups program of Telefónica resulting one of its winners and continued with its development in the BIC Gipuzkoa accelerator. Oscar also completed an internship at Mediatek in Cambridge (England) as an application developer. As academic merits, Oscar received the award for the best academic record of his promotion and honors in his TFM.

Ekhi Zugasti graduated in telecommunications engineering from Mondragon Unibertsitea in 2009 and received his PhD in the area of civil engineering knowledge from the Polytechnic University of Catalonia (UPC) in 2014. The title of his thesis is "Design and validation of a methodology for wind energy structures health monitoring". In 2017 he joined the Research Group of Intelligent Systems for Industrial Systems of Mondragon Unibertsitea, where he designs, develops and implements projects related to data analysis, the Big Data paradigm and anomaly detection. Previously, he developed his professional career in the IK4-Ikerlan technology centre, first in the Sensor Department, where he developed projects within the ETICS line, until December 2015, in the Mechanics Department, where he designed and developed projects related to Data Analysis for monitoring the condition of machinery parts, and finally, until 2017, he finished in the Control and Monitoring Department, in projects for monitoring and data analysis in Advanced Manufacturing problems.

Jon Rodriguez born in Donostia on 07/14/1990, graduated in Computer Engineering at the University of Mondragon (2008-2012). He is currently working as a Research Manager at Koniker, a position in which he performs technical and management tasks (administrative and people). In the present project he works leading the tasks of the Koniker staff, defining the use cases and giving technical support whenever necessary.

Urko Zurutuza is the principal investigator of the Intelligent Systems for Industrial Systems Research Group, accredited by the Basque Government as an A type group (max. qualification of excellence). Urko is a Technical Engineer in Systems Computing, a Higher Engineer in Computing, and a Doctor in Computing from Mondragon Unibertsitatea. He has participated in more than 50 publicly funded research projects, is the author of more than 45 publications in academic conferences and journals, and is a member of more than 20 scientific committees of international conferences, such as DIMVA (SIG SIDAR Conference on Detection of Intrusions and Malware & Vulnerability Assessment) or CRITIS (International Conference on Critical Information Infrastructures Security). Urko has supervised 7 PhD thesis, and has 4 ongoing. He has been the General Chair in conferences such as RECSI 2012, DIMVA 2016, JNIC 2018, and recent RAID 2020 and 2021. Urko is a member of the Steering Board of RENIC (National Network of Excellence in Cybersecurity Research), and part of the National Forum on Cibersecurity. He has coordinated research projects at regional, national and European level, has experience in FP6, FP7, H2020, Joint Programmes (JTI's) such as Artemis and Ambient Assisted Living (AAL), and has been the main coordinator of the recently completed European project MANTIS (47 partners, 30M budget).