SN Computer Science (2024) 5:371
https://doi.org/10.1007/s42979-024-02635-5

ORIGINAL RESEARCH

Recognizing Bearings' Degradation Stage Using Multimodal Autoencoder to Learn Features from Different Time Series

Antonio Luca Alfeo[1,2] · **Mario G. C. A. Cimino**[1,2] · **Guido Gagliardi**[1,3,4]

Received: 14 April 2023 / Accepted: 17 January 2024

Abstract

Utilizing machine learning technologies to monitor assets' health conditions can improve the effectiveness of maintenance activities. However, accurately recognizing the current health degradation stages of industrial assets requires a time-consuming manual feature extraction due to the wide range of observable measures (e.g., temperature, vibration) and behaviors characterizing assets' degradation. To address this issue, feature learning technology can transform minimally processed time series into informative features, i.e., able to simplify the classification task (e.g., recognizing degradation stages) regardless of the specific machine learning classifier employed. In this work, minimally preprocessed time series of vibration and temperature of industrial bearings are exploited by an autoencoder-based architecture to extract degradation-representative features to be used for recognizing their degradation stages. Different autoencoder architectures are employed to compare their data fusion strategies. The effectiveness of the proposed approach is evaluated in terms of recognition performance and the quality of the learned features by using a publicly available real-world dataset and comparing the proposed approach against a state-of-the-art feature learning technology. We tested three different multimodal autoencoder-based feature learning approaches, i.e., shared-input autoencoder (SAE), multimodal autoencoder (MMAE), and partition-based autoencoder (PAE). All the AE-based architecture results in classification performances greater or comparable with the state-of-the-art feature learning technology, despite being trained in an unsupervised fashion. Also, the features provided via PAE correspond to the greatest performances in recognizing bearings' degradation stage, providing high-quality features both from a classification and clustering perspective. Unsupervised feature learning methodologies based on multimodal autoencoders are capable of learning high-quality features. These result in greater degradation stages recognition performances when compared to supervised state-of-the-art feature learning technology. Also, this enables the correct representation of the expected progressive degradation of the bearing.

Keywords Deep feature learning · Data fusion · Multimodal autoencoder · Contrastive learning · Predictive maintenance · Smart manufacturing

Introduction

According to the Industry 4.0 paradigm, industrial processes can be remarkably improved by using machine learning [1–3]. For instance, machine learning techniques can be employed to provide the so-called predictive maintenance (PdM) [4]. PdM aims at assessing the current degradation state of industrial assets to perform maintenance operations just before the breaking point. If compared to reactive maintenance (i.e., fix after a failure) and preventive maintenance (fix periodically), PdM allows for avoiding failures while fully exploiting the whole remaining useful life (RUL) of an asset's component [5].

Since RULs are greatly influenced by asset usage while in an unhealthy stage, its prediction is often unreliable. Thus, degradation stage estimations are often preferred RUL estimates in real-world applications [6]. A trade-off between interpretability and complexity determines the number of stages used to characterize the degradation process. For instance, a few easy-to-interpret stages can be used to describe a degradation process that is consistent and progressive. Most of the research works [6] employ

This article is part of the topical collection "Innovative Intelligent Industrial Production and Logistics 2022" guest edited by Alexander Smirnov, Kurosh Madani, Hervé Panetto and Georg Weichhart.

Extended author information available on the last page of the article

Published online: 29 March 2024

three [7], four [8], or even five stages [9] to characterize the degradation process.

Moreover, PdM approaches need to be provided with some features (e.g., statistical measures) extracted from some measurements (e.g., temperature, vibration, acoustic noise) that need to be informative about the degradation process. However, due to the many measures that can be taken into account, and the diversity of degradation processes across industries and machines, it is difficult to have a feature extraction process that is generalizable across various PdM applications [10]. As a result, choosing and transforming such measurements into informative features requires intensive and time-consuming collaborations between data scientists and maintenance experts.

Thus, more automatic and adaptive feature extraction processes are required in the PdM context [11]. Those can be obtained by using feature learning technology [12]. *Feature learning* approaches automatically transform minimally processed data into informative features aimed at simplifying the classification tasks [13, 14]. Prior domain knowledge, such as which features to include or exclude for the analysis of a specific measure, is not required with feature learning.

This is especially convenient when employing multiple and heterogeneous sources [6], which would require a specific preprocessing and feature extraction for each one of them. The need for multimodal approaches for PdM is indeed emphasized in different recent surveys such as [15].

In this context, deep learning approaches can provide a higher-level representation of the inputs that can be used as features in a classification problem. Moreover, by being characterized by hierarchically stacked nonlinear modules, deep learning approaches allow the processing of data from different modalities simultaneously to provide some sort of information fusion. Indeed, many multimodal feature learning approaches are implemented via deep learning technology [16], and especially via deep autoencoders (AE) [10].

This study compares different unsupervised AE-based architectures for multimodal feature learning, each one implementing a different data fusion strategy. The quality of the learned features and degradation stage recognition performances are also compared against the classic feature extraction process and the state-of-the-art technology in supervised feature learning. The proposed approach has been tested on a well-known PdM benchmark dataset consisting of three real-world cases study addressing the degradation of industrial bearings.

The paper is structured as follows. In section 2, the literature review is presented. Section 3 details the proposed approach. The case study and the experimental setup are presented in Sect. 4. Finally, Sect. 5 and 6 discuss the obtained results and the conclusions, respectively.

Related Works

This section presents a survey of the state-of-the-art addressing feature learning approaches.

Principal component analysis (PCA) and linear discriminant analysis (LDA) can be considered the first feature learning algorithms [16], and were originally designed for dimensionality reduction. The first approaches able to map the data into a higher dimensional space was the kernel version of those linear dimensionality reduction algorithms, i.e., kernel PCA (KPCA) [17] and generalized discriminant analysis (GDA) [18], which are the kernel version of PCA and LDA, respectively. Those feature learning algorithms, either linear or nonlinear, belong to the shallow learning paradigm. Since the early 2000s, many deep learning approaches have been proposed to learn informative data representations. Those can result in better abstractions of the original data for subsequent classification tasks compared to shallow learning approaches [19].

By consisting of a stack of layers of artificial neurons, deep learning architecture intrinsically distill high-level information (i.e., features) by performing nonlinear combinations of the input data [20, 21]. Specifically, the inputs provided to the architecture are processed by each layer to the next one. The intermediate information representations between two layers (i.e., the so-called latent space) can also be used as features in the following recognition task [22].

In this context, generative adversarial networks (GANs) and autoencoders (AEs) are among the most used deep learning architecture for feature learning [23].

GANs [24] were originally designed for data generation and are made of two neural networks: a generator (G) and a discriminator (D). G is not informed about the distribution of the real data and aims to generate fake data to fool D. D aims to discriminate fake data generated by G from the real ones. Once trained, G can be used for data generation. Recently, GANs and their variants are also used for feature learning [25]. For instance, BiGAN [26] is a GAN specifically designed to learn the latent representation of the data. Unfortunately, the use of random noise as input for the G network makes GAN's learning projection to be unpredictable [27].

An AE consists of two neural networks trained in an end-to-end fashion: the *encoder* works as a bottleneck to obtain a compact representation of the inputs, whereas the *decoder* reconstructs the input data using such a compact representation. By using a loss function aimed at maximizing the similarity between its input and output, the AE does not need any labeled data to be trained and thus it is an unsupervised approach. By embedding

enough information to reconstruct the whole input, the compact representations provided by the trained encoder are considered informative enough to be used as features for classification tasks. Together with this feature learning capability, some autoencoder architecture can also fuse multisensory data [28]. Specifically, AE-based approaches can provide multimodal-data fusion at three different levels [22]:

- At the *data level*, the AE processes the concatenation of the original input data for each modality and provides a single multimodal representation for both of them [29].
- At the *architecture level*, the AE processes the input of each modality independently, but the last layers of the encoder are shared among different modalities and thus provide a single multimodal representation [30].
- At the *representation level*, there are two independent AEs processing the input of each modality. The obtained representations are then concatenated to obtain a single multimodal representation [31].

For instance, in [32], different multimodal AE approaches for handling both audio and video inputs are compared. In this context, the so-called shared modality AE concatenates multimodal features as input and reconstructs those together (data-level fusion), whereas the multimodal AE consists of a multi-input–multi-output network (architecture-level fusion), in which each modality is provided and reconstructed separately while being processed together by the network. As emerged from the analysis in [33], the capability of handling and fusing different modalities while providing feature learning is highly required to improve the recognition performances, especially in a fault detection scenario. In this regard, the authors in [34] propose a deep coupling autoencoder (DCAE) to process vibration and acoustic data to obtain a multimodal representation to be used for fault diagnosis. Specifically, a coupling autoencoder (CAE) is constructed to couple the hidden representations of two single-modal autoencoders obtaining a joint representation between different multimodal sensory data, and then a DCAE model is devised for learning the joint higher-level feature.

Alternatively to unsupervised feature learning approaches based on AE, a neural network can provide feature learning also in a supervised fashion, employing specifically designed loss functions. For instance, the multi-similarity loss is aimed at learning a higher-level data representation in the latent space of a neural network by maximizing the separability of the learned representations clustered for the target classes. This approach can be exploited to learn features that can be used to recognize the degradation state of an industrial component from its time series [3]. An implementation of the multi-similarity loss is provided by tensorflow similarity [35] and represents the state-of-the-art learning features for similarity ranking problems. In the following, the feature extraction approach based on multi-similarity loss will be referred to as the similarity-based encoder.

Design

In this section, the design of the proposed approach is detailed. It consists of three functional modules, i.e., data preparation, feature extraction, and degradation stage classification (Fig. 1).

Both the vibration and temperature time series are minimally pre-processed via the *data preparation module*. Firstly, each time series is segmented and associated with a degradation stage label (more on the labeling of each segment in Sect. 4). To do so, 30 s semi-overlapping time windows are employed as a reference for the segmentation process. Unlike the temperature, the vibration is characterized by a strong fluctuating behavior. Thus, its informativeness is typically extracted in the frequency domain rather than in the time domain [36, 37]. For this reason, the *data preparation module* processes the segments of the vibration time series by transforming those via the discrete Fourier transform evaluated computing the fast Fourier transform with N equal to the length of the input signal; then the real and the imaginary parts of the signal were stored; also the probability density function and the kurtosis of each input signal were evaluated. Following a model centric approach, no further assumption has been made about the range of informative frequencies. In fact, the tested features learning algorithm (SE, AE, PAE, MAE) share the ability to autonomously learn the informative part of the input data [13]. Doing this, the system is in charge, during training, to find and exploit such information allowing it to be suitable for different types of bearings with different frequency failure rates.

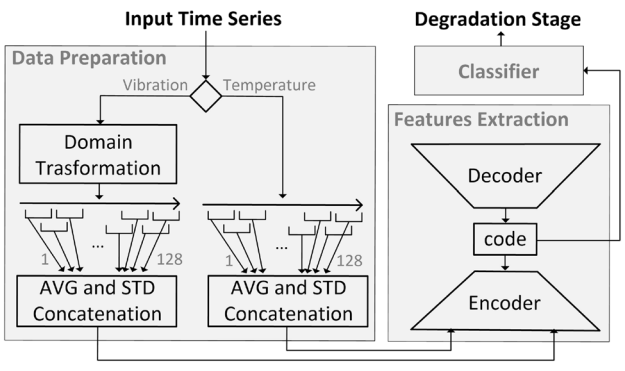

Fig. 1 Architecture of the proposed approach consisting of three functional modules

The segments of the temperature time series and the discrete Fourier transform of the vibration segment are treated as numerical arrays and split into 128 semi-overlapped sub-parts each. For each sub-part, we compute the average and standard deviation. Then we concatenate and rescale their values between 0 and 1 via a min–max procedure. In essence, the *data preparation module* provides four numerical arrays of 128 elements for each 30 s observation: two arrays are obtained via the discrete Fourier transform of the vibration signal and two via the temperature one.

The *feature extraction module* employs an approach based on AEs to process the output of the *data preparation module* and learn degradation-representative features. As introduced in Sect. 2, AE-based architectures can provide different data fusion strategies. Specifically:

- the *shared-input autoencoder* (SAE) provides data-level fusion, i.e., concatenates the input of each modality, and then processes them via an autoencoder to learn a multimodal representation (Fig. 2.a);
- the *multimodal autoencoder* (MMAE) provides architecture-level data fusion, i.e., the input of each modality feeds a distinct part of the AE'S neural network; the multimodal representation is obtained by combining the processing of the inputs via some shared layers of neurons (Fig. 2.b);
- the *partition-based autoencoder* (PAE) provides representation-level data fusion, i.e., processes the input for each modality via different autoencoders and concatenate the representations obtained from each one of them to have a multimodal representation (Fig. 2.c).

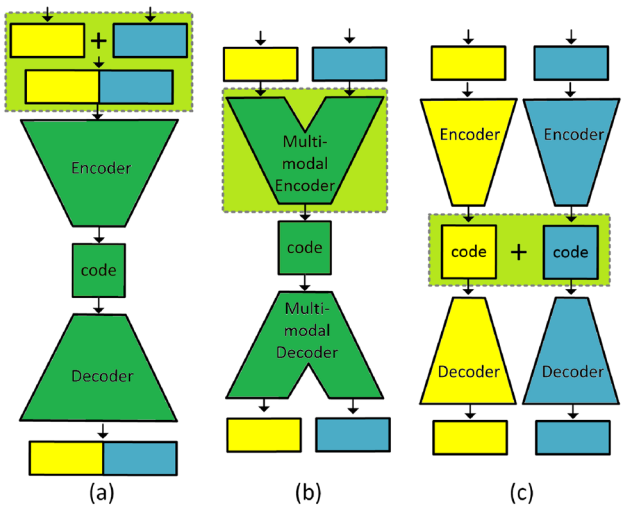

Fig. 2 Autoencoder-based multimodal feature learning approaches, i.e., shared-input autoencoder (**a**), multimodal autoencoder (**b**), and partition-based autoencoder (**c**). In blue and yellow are the parts of the autoencoder that work with one single modality. The dashed box highlights the modalities-fusion phase

Figure 2 exemplifies the above-described multimodal feature learning strategies. The processing of the inputs of each modality is colored in yellow or blue. The dashed box highlights the modalities fusion phase. In dark green, we represent the AE components that work in a multimodal fashion.

Once trained, the *feature extraction module* can provide the codes (or their concatenation) as a multimodal representation of the inputs for each modality. Such representation will be used as a feature for the *degradation stage classifier*.

As specified in Sect. 1, to test this capability, the proposed approach uses a number of different classifiers, as provided by the well-known Python library *scikit − learn* [38].

Experimental Setup

In this section, the experimental dataset and the experimental setup are described. This is used for the evaluation of the effectiveness of the proposed approach (Fig. 3).

In our experiments, we employ a publicly available dataset obtained via the experimental platform Pronostia [39]. The platform provides the progressive degradation of real-world industrial bearings and collects the time series of vibration (25.6 kHz) and temperature (10 Hz) during the degradation process. The Pronostia dataset comprises three distinct cases of study denoted as B11, B12, and B21 [39], each corresponding to different bearings (indicated by the second number in the case of study name) and bearing operating conditions. The initial digit in the case of study name delineates the operating condition of the bearing: B1X pertains to conditions between 1800 rpm and 4000 N, while B2X relates to conditions between 1650 rpm and 4200 N.

The time series are segmented into semi-overlapping time windows with a duration of 30 s, and associated with the corresponding degradation stage label. In this study, three

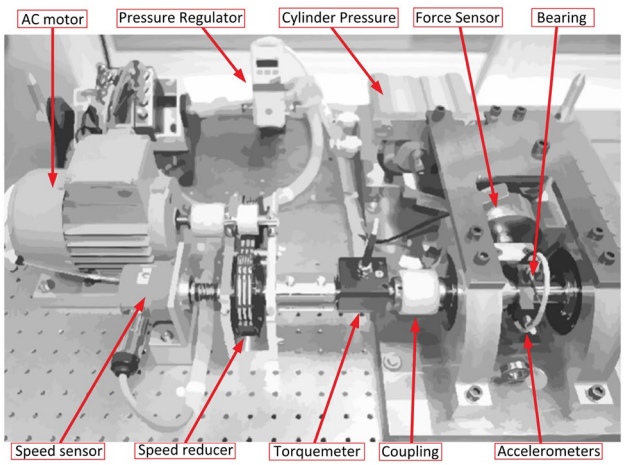

Fig. 3 The Pronostia plaform [39]

degradation stages are taken into account: regular, degraded, and critical. To determine the time points at which the degradation stage shifts, the vibration time series is examined. Specifically, the instant in which the vibration results are consistently equal to or greater than 1 g is considered as the transition between regular and degraded health stages [31]. When in the degraded health stage, the instant in which the root mean square of the vibration suddenly increases is considered the transition from the degraded to the critical health stage [40]. More details about this labeling procedure are provided in [31]. In Table 1, we report the resulting number of time series segments (and so instances) for each case study and degradation stage.

As per the results in [3], all the AE-based architectures used in this study are characterized by the mean absolute error as training loss, 128 as *batch size*, *Relu* as activation function, *Adam* as an optimization algorithm, and a symmetric decoder and encoder. It means that their neural networks consist of the same number of layers and inverted layers' order. The encoder (decoder) features four layers consisting of 128, 64, 32, and 16 (16, 32, 64, and 128) artificial neurons, respectively. The multimodal encoder features the same number of layers and neurons for each modality, except for the most internal layer (i.e., the one with 16 neurons). This layer is indeed replaced with three layers (consisting of 64, 32, and 16 neurons) shared among both modalities. The number of neurons of the input (output) layer of the encoder (decoder) varies to fit the input length, e.g., SAE's input is twice as long as PAE's one. This allows us to have a comparable number of trainable parameters for each AE-based feature extraction module.

The proposed approach is compared to a state-of-the-art feature learning approach, i.e., the multi-similarity loss. As mentioned in Sect. 2, in September 2021 the implementation of the multi-similarity loss was released by Google via the package *Tensorflow Similarity*. Specifically, the loss function provided by *Tensorflow Similarity* considers the similarity, measured as the inverse of the Euclidean distance, between the representation of three data points in the latent space, i.e., the anchor (A), the positive (P), and the negative (N). P (N) is chosen among the samples in the batch characterized by the same (different) class with respect to A. The neural network is trained to progressively reduce the distances between A and P, and increase the distance between A and N, resulting in a difference between these two distances greater than a given margin for all the training samples. Unlike AE-based approaches, this feature learning approach is supervised and specifically designed to disentangle instances of different classes in the latent space. This should correspond to an improved performance for the subsequent recognition task, at the cost of increased training time, as demonstrated in [3].

Table 1 Instances per class and case study

Degradation stage	B11	B12	B21
Regular	1871	748	753
Degraded	1665	319	371
Critical	181	73	74

As evident from Table 1, the classes in our degradation stage classification problem are unbalanced, i.e., the more severe the degradation stage, the fewer are the instances in the dataset. For this reason, the classification performance is measured in terms of *F1-score* [41], i.e., the harmonic mean of precision and recall (Eq. 1).

Given one class to recognize, the precision is the ratio between the number of true positives (i.e., samples correctly recognized as that class) and the number of all positives (i.e., all the samples recognized as that class). The recall, instead, is the ratio between the number of true positives and the sum of true positives and false negatives, i.e., the number of all samples that should have been identified as belonging to that class. Since our classification problem features three different classes, the average F1-score among all the classes (i.e., the global F1 score) is considered the main recognition performance measure. The F1-score is bounded between 0 and 1. An F1-score equal to 1 means that there are no false positives (e.g., a critical stage recognized as a regular one) and false negatives (e.g., a regular stage recognized as a critical one). An F1-score equal to 0 means that either the precision or the recall is zero, i.e., there are no true positives (e.g., a correctly recognized degradation stage). For the sake of readability, the F1-score is presented as a percentage, i.e., bounded between 0 % (worst case) and 100% (best case).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}. \quad (1)$$

To test if the learned features are informative regardless of the machine learning approach used for the classification, in our experimentation we employ six different classifiers provided by $scikit-learn$, specifically:

- *KNeighborsClassifier* [3] (KNN), an ML classifier that determines the class for a new sample according to the class of a given number of closest training samples.
- *LinearDiscriminantAnalysis* [42] (LDA), an ML classifier that employs Bayes' rule and class conditional densities to determine a linear decision boundary to separate samples in different classes.
- *Support vector machine* [43](SVM), a kernel-based ML classifier aimed at predicting highly calibrated class membership probabilities.
- *ExtraTreesClassifier* [44] (ET), *GradientBoostingClassifier* [43] (GB), and *RandomForestClassifier* [45] (RF), three ML classifiers based on ensembles of decision trees

that are well known for their fast convergence and great classification performances.

A feature can be considered informative if it eases the separation of the instances among classes, thus improving the performance of the classification task [46]. In this regard, clustering quality metrics can be employed to measure how well the learned features space separates different classes. Indeed, previous studies such as [47] and [48] find that such separability may actually correlate with the final classification accuracy.

In this context, the so-called *silhouette coefficient*, or silhouette score, quantifies how similar an object is to its own cluster compared to other clusters. The silhouette score ranges $[-1, +1]$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. For a given sample $i \in C_I$, where C_1, $C_2 ... C_N \in D$ are the sets of different clusters in the dataset D; the silhouette score of i can be computed as:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}. \tag{2}$$

In Eq. 2, $a(i)$ is the mean distance between i and all the elements of its cluster C_I, whereas $b(i)$ is the smallest mean distance of i to all the elements in any other cluster. In our study, each class (e.g., degradation stage) is made to correspond to a cluster in the silhouette score calculation.

$$\begin{aligned} a(i) &= \frac{1}{\|C_I\| - 1} \sum_{j \in C_I, i \neq j} d(i,j) \\ b(i) &= \min_{J \neq I} \frac{1}{\|C_J\|} \sum_{j \in C_J} d(i,j) \end{aligned}. \tag{3}$$

We also provide visualizations of the instances in the learned feature space to qualitatively evaluate the class separability. Due to the multidimensionality of the learned feature space, it is challenging to graphically represent it. Thus, to this aim, we employ the projections along the three principal components' directions. Although this projection is useful for qualitative analysis and visualization, it may not fully capture the actual closeness between the instances in the feature space [49].

Our experimental results are presented as average socres obtained via a stratified Monte Carlo ten cross-fold validation schema.

Results

In the following, the different feature extraction approaches are shortened as follows: partition-based autoencoder (PAE), shared-input autoencoder (SAE), multimodal autoencoder (MMAE), similarity-based encoder (SE).

We evaluated the quality of the features learned by our unsupervised autoencoder-based architectures (PAE, SAE, and MAE) in comparison with the supervised one (SE) by showing the classification perfomances (F1-score) of different ML classifiers which take as input the features learned by the different types of encoders, in the recognition of the degradation stages "regular", "degraded", and "critical". These results are shown in Table 2.

The effectiveness of the proposed architecture was also evaluated on a more complex classification task, i.e., with a larger number of degradation stages. To do so, the degradation stages "regular" and "degraded" were split (i.e., considering half of their duration) into two stages each, resulting in a five-stage bearing degradation recognition. The classification performance achieved with each variant of the proposed approach is documented in Table 2.

As per results in Table 2, the combination of the boosting-based approaches (i.e., ET, GB and RF) with PAE learned features outperform the non-boosting based approaches (i.e., KNN, LDA and SVM). In particular for the B11 case of study, GB achieves 99,44% F1-score in three degradation stage classification, while RF achieves 98,44% for five-stage classification. For B12, ET achieves 95,25% F1-score in three degradation stage classification and 92,54% in five-stage classification. For B21, GB and RF achieve 99,17% F1-score in three degradation stage classification using PAE's features, while ET achieves 96,08% for five-stage classification.

Moreover the boosting-based approaches share better overall performances if we consider the features learned by SAE, PAE and MMAE, with respect to the other ML classifiers, and the performances are consistent with the one provided via SE. This result can be explained considering the fact that KNN, SVM and LDA are classifiers whose performances are highly impacted by the spatial class separability of the features in input. This class separability is directly maximized by SE providing generally better results for these types of classifiers.

This result is confirmed by Fig. 4 in which the projections over the three principal components obtained from the features learned by each encoder are visualized in a 3D plot. In the figure, it is possible to see how SE results in clusters of data for each degradation stage characterized by a clear separation, and hence the multi-similarity loss maximizes their class separability.

Table 2 Average % F1-score obtained for all the cases of the study (B11, B12, and B21) with multiple approaches for the three- and five-class multiclass classification problem

		3 degradation stages				5 degradation stages			
Case study	ML classifier	SAE (%)	PAE (%)	MMAE (%)	SE (%)	SAE (%)	PAE (%)	MMAE (%)	SE (%)
B11	ExtraTreesClassifier	99.01	**99.06**	98.79	98.60	97.66	98.41	97.20	95.99
	GradientBoostingClassifier	98.76	99.44	98.63	98.66	97.20	98.28	97.20	95.86
	RandomForestClassifier	98.63	99.11	98.68	98.60	97.39	**98.44**	97.12	95.99
	KNeighborsClassifier	98.55	97.10	98.36	98.74	94.81	89.95	94.76	96.21
	LinearDiscriminantAnalysis	97.90	98.20	97.18	98.71	92.20	90.40	92.04	96.16
	SVM	98.12	97.53	98.44	98.76	94.14	89.57	93.52	96.05
B12	ExtraTreesClassifier	93.42	**95.35**	92.89	92.37	89.04	**92.54**	89.21	86.23
	GradientBoostingClassifier	93.16	95.00	92.81	92.28	88.42	91.58	88.33	87.11
	RandomForestClassifier	93.51	94.56	92.89	93.16	87.72	92.11	88.68	87.28
	KNeighborsClassifier	88.68	87.63	90.53	92.46	80.96	79.47	83.25	86.93
	LinearDiscriminantAnalysis	88.07	89.39	86.75	93.07	75.53	78.77	75.70	87.89
	SVM	89.21	91.84	88.68	92.54	76.84	75.26	77.19	88.25
B21	ExtraTreesClassifier	96.42	99.00	96.42	94.92	90.25	**96.08**	90.00	87.00
	GradientBoostingClassifier	96.08	**99.17**	95.75	94.42	89.25	95.42	89.42	87.75
	RandomForestClassifier	96.33	**99.17**	95.92	94.58	89.58	96.00	89.92	87.42
	KNeighborsClassifier	93.75	83.50	93.50	94.75	84.50	69.67	83.92	87.58
	LinearDiscriminantAnalysis	87.83	93.00	87.50	94.75	81.08	82.58	80.67	87.83
	SVM	89.92	89.83	90.25	95.00	80.83	78.00	81.75	87.50

In bold we report the best performance for each case study and number of degradation stages

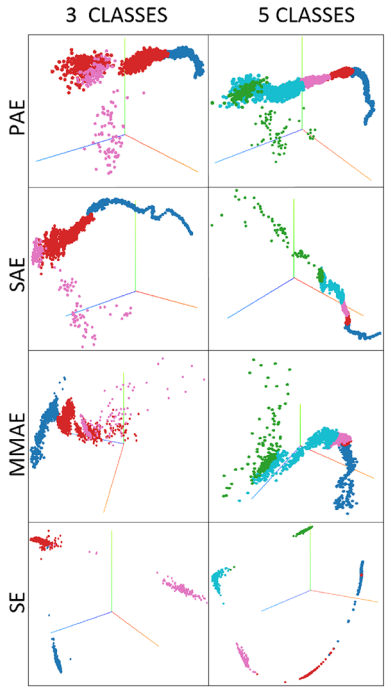

Fig. 4 Features learned with the B11 case study and projected over three principal components. The degradation stages range from regular (blue) to critical (pink with 3 classes, green with 5 classes)

The fact that approaches based on boosting using the features learned by SAE, PAE, and MMAE outperform the approaches with features learned by SE implies that SE learns features less informative from the classification perspective than the others approaches. Moreover, as Fig. 4 shows, SE learns a representation that is highly separated and compacted. This representation does not resemble the expected distribution of the feature, which we expect to gradually shift from one degradation stage to another, as evident from the principal components of the features learned by MMAE and SAE. The PCA projection space depicted in Fig. 4 employs the first three principal components derived from the projected data with case study B11. The projections learned by the SE model exhibit a more distinct separation between the classes. This visualization does not necessarily imply that the other (less clearly separated) PCA projection spaces, obtained through SAE, PAE, and AE, should always correspond to significantly worse classification performance. Indeed, those recognition performances are also due to the predictive capability of the ML model. In fact, the ML approaches that mostly rely only on the spatial distribution of the data such as KNN, SVM, and LDA result in better classification performances using SE-learned features with respect to the ones obtained via SAE, PAE, and AE. The same result can be observed considering the other study cases (B21 and B22). On the other hand, models with a greater prediction capability, such as decision tree

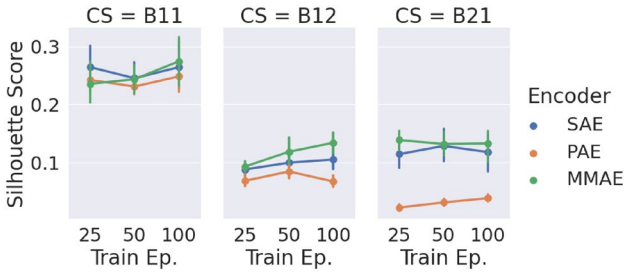

Fig. 5 Silhouette score (average and standard deviation) comparison considering different case studies (B11, B12, and B21) varying the number of training epochs

ensembles, result in comparable classification performances when considering the features learned by SAE, PAE, AE, and SE.

Considering the comparison between SAE, PAE and MMAE, the features learned by PAE achieve always the best classification performances; on the other hand, as discussed in [3], the PAE architecture needs to train one autencoder module for each modality involved in the classification task, and hence needs much more time to be trained properly. For this reason there are use cases where the usage of MMAE and SAE should be taken into consideration, since they provide classification results completely in line with PAE, but they require much less training time.

For this reason, we compared the features learned by MMAE, SAE, and PAE also considering clustering metrics, i.e., the silhouette score, and show their results in Fig. 5. We computed the silhouette score for each encoder as the mean silhouette score for each train feature learned considering as cluster identifier the degradation stage of the sample. Hence, the silhouette score shows how much the degradation stage clusters are separated in the encoding space by the architectures. As shown in Fig. 5, PAE and SAE share generally better silhouette scores over the training epochs and case of study (B11, B12, and B21) with respect to MMAE.

Conclusion

The proposed architecture employs and compares different multimodal AEs to extract representative features from different minimally processed time series data. Specifically, the vibration and temperature are used to detect the degradation stage of an industrial bearing. A publicly available real-world dataset is employed to evaluate the effectiveness of the proposed approach against state-of-the-art technology in feature learning. The results indicate that using unsupervised features learning methodologies, such as shared-input autoencoder (SAE), multimodal autoencoder (MMAE), and partition-based autoencoder (PAE) results in high-quality learned features that can easily differentiate between different classes despite the ML classifiers employed, and especially if the ML classifier is based on ensembles of decision trees, such as *ExtraTreesClassifier*, *GradientBoostingClassifier*, and *RandomForestClassifier*.

Moreover, these autoencoder architectures achieve average recognition performances that are higher or comparable with those achieved by employing state-of-the-art techniques (specifically, multi-similarity loss) for feature learning even though they are trained in an unsupervised fashion and despite the number of degradation stages taken into account.

The promising results achieved in our study confirm how the PAE architecture learns superior quality features with respect to the other approaches at the expense of greater training time which also is negatively impacted by the number of modalities involved [3]. For this reason, SAE and MMAE architectures should also be taken into consideration, since they can learn high-quality features from the data (with results in line with the ones learned by PAE), with less training time. Moreover, for bearing degradation stage recognition, SAE should be preferred to MMAE, since it usually learns features that are more separable with respect to the degradation stage, considering different use cases and training epochs.

Acknowledgements The work was partially supported by: (i) the University of Pisa, in the framework of the PRA 2022 101 project "Decision Support Systems for territorial networks for managing ecosystem services"; (ii) the European Commission under the NextGenerationEU program, Partenariato Esteso PNRR PE1—"FAIR—Future Artificial Intelligence Research"-Spoke 1 "Human-centered AI"; (iii) the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence), in the framework of the "Reasoning" project, PRIN 2020 LS Programme, Project number 2493 04-11-2021, and in the framework of the project "OCAX—Oral CAncer eXplained by DL-enhanced case-based classification" PRIN 2022 code P2022KMWX3. Thw work was partly funded by the European Commission under the Next Generation EU programme, PNRR-M4 C2, Investment 1.5 "Creating and strengthening of "innovation ecosystems", building "territorial R &D leaders", project "THE Tuscany Health Ecosystem", Spoke 6 "Precision Medicine and Personalized Healthcare". The work was partially funded by the European Union-Next Generation EU (National Sustainable Mobility Center CN00000023, Italian Ministry of University and Research Decree n. 1033-17/06/2022, Spoke 10)".

Funding Open access funding provided by Università di Pisa within the CRUI-CARE Agreement.

Data availability All data analyzed in this study are publicly available and included in the published articles [39].

Declarations

Conflict of interest The authors declare that they have no conflict of interest and that an ethical statement is not applicable to this research.

References

1. Alfeo AL, Cimino MG, Vaglini G. Technological troubleshooting based on sentence embedding with deep transformers. J Intell Manuf. 2021;32(6):1699–710.
2. Alfeo AL, Cimino MG, Gagliardi G. Concept-wise granular computing for explainable artificial intelligence. Granul Comput. 2023;8(4):827–38.
3. Alfeo AL, Cimino M, Gagliardi G. Automatic feature extraction for bearings' degradation assessment using minimally pre-processed time series and multi-modal feature learning. In: Proceedings of the 3rd International Conference on Innovative Intelligent Industrial Production and Logistics (IN4PL 2022); 2022.
4. Jimenez JJM, Schwartz S, Vingerhoeds R, Grabot B, Salaün M. Towards multi-model approaches to predictive maintenance: a systematic literature survey on diagnostics and prognostics. J Manuf Syst. 2020;56:539–57.
5. Wan J, Tang S, Li D, Wang S, Liu C, Abbas H, et al. A manufacturing big data solution for active preventive maintenance. IEEE Trans Industr Inf. 2017;13:2039–47.
6. Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: a systematic review from data acquisition to RUL prediction. Mech Syst Signal Process. 2018;104:799–834.
7. Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: Proceedings of the 26th annual international conference on machine learning, 2009. p. 1073–80.
8. Scanlon P, Kavanagh DF, Boland FM. Residual life prediction of rotating machines using acoustic noise signals. IEEE Trans Instrum Meas. 2012;62:95–108.
9. Kimotho JK, Sondermann-Wölke C, Meyer T, Sextro W. Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution–particle swarm optimization. Chem Eng Trans. 2013;33:619–24.
10. Ran Y, Zhou X, Lin P, Wen Y, Deng R. A survey of predictive maintenance: Systems, purposes and approaches. arXiv preprint arXiv:1912.07383. 2019;.
11. Yan W, Yu L. On accurate and reliable anomaly detection for gas turbine combustors: a deep learning approach. In: Annual conference of the PHM society. vol. 7; 2015.
12. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013;35:1798–828.
13. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA, Bottou L. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res. 2010;11:3371–408.
14. Gagliardi G, Alfeo AL, Catrambone V, Cimino MG, De Vos M, Using Valenza G, Learning contrastive, to inject domain-knowledge into neural networks for recognizing emotions. In,. IEEE symposium series on computational intelligence (SSCI). IEEE. 2023;2023:1587–92.
15. Merkt O. Predictive models for maintenance optimization: an analytical literature survey of industrial maintenance strategies. Information Technology for Management: Current Research and Future Directions; 2019. p. 135–54.
16. Zhong G, Ling X, Wang LN. From shallow feature learning to deep learning: benefits from the width and depth of deep architectures. Wiley Interdiscip Rev: Data Min Knowl Discov. 2019;9: e1255.
17. Schölkopf B, Smola A, Müller KR. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 1998;10(5):1299–319.
18. Baudat G, Anouar F. Generalized discriminant analysis using a kernel approach. Neural Comput. 2000;12(10):2385–404.
19. Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, et al. Decaf: A deep convolutional activation feature for generic visual recognition. In: International conference on machine learning. PMLR; 2014. p. 647–655.
20. Tang S, Yuan S, Zhu Y. Deep learning-based intelligent fault diagnosis methods toward rotating machinery. IEEE Access. 2019;8:9335–46.
21. Gagliardi G, Alfeo AL, Catrambone V, Candia-Rivera D, Cimino MG, Valenza G. Improving emotion recognition systems by exploiting the spatial information of EEG sensors. IEEE Access. 2023;11:39544–54.
22. Gao J, Li P, Chen Z, Zhang J. A survey on deep learning for multimodal data fusion. Neural Comput. 2020;32(5):829–64.
23. Deng L. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans Signal Inf Process. 2014;3: e2.
24. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. Advances in neural information processing systems. 2014;27.
25. Hong Y, Hwang U, Yoo J, Yoon S. How generative adversarial networks and their variants work: an overview. ACM Comput Surv (CSUR). 2019;52(1):1–43.
26. Donahue J, Krähenbühl P, Darrell T. Adversarial Feature Learning. In: International conference on learning representations.
27. Suryawati E, Pardede HF, Zilvan V, Ramdan A, Krisnandi D, Heryana A, et al. Unsupervised feature learning-based encoder and adversarial networks. J Big Data. 2021;8(1):1–17.
28. Yan X, Liu Y, Jia M. Health condition identification for rolling bearing using a multi-domain indicator-based optimized stacked denoising autoencoder. Struct Health Monit. 2020;19:1602–26.
29. Gecgel O, Ekwaro-Osire S, Gulbulak U, Morais TS. Deep convolutional neural network framework for diagnostics of planetary gearboxes under dynamic loading with feature-level data fusion. J Vib Acoust. 2022;144(3): 031003.
30. Shin B, Lee J, Han S, Park CS. A study of anomaly detection for ICT infrastructure using conditional multimodal autoencoder. J Intell Inf Syst. 2021;27(3):57–73.
31. Alfeo AL, Cimino MG, Vaglini G. Degradation stage classification via interpretable feature learning. J Manuf Syst. 2022;62:972–83.
32. Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY. Multimodal deep learning. In: ICML; 2011.
33. Qian J, Song Z, Yao Y, Zhu Z, Zhang X. A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. Chemom Intell Lab Syst. 2022;p. 104711.
34. Ma M, Sun C, Chen X. Deep coupling autoencoder for fault diagnosis with multimodal sensory data. IEEE Trans Industr Inf. 2018;14(3):1137–45.
35. Elie Bursztein SLOVFC James Long. TensorFlow similarity: a usable, high-performance metric learning library. Fixme. 2021.
36. Pandarakone SE, Masuko M, Mizuno Y, Nakamura H, Deep neural network based bearing fault diagnosis of induction motor using fast Fourier transform analysis. In,. IEEE energy conversion congress and exposition (ECCE). IEEE. 2018;2018:3214–21.
37. Gagliardi G, Alfeo AL, Catrambone V, Cimino MG, De Vos M, Valenzal G. Fine-grained emotion recognition using brain-heart

interplay measurements and eXplainable convolutional neural networks. In: 2023 11th international IEEE/EMBS conference on neural engineering (NER). IEEE; 2023. p. 1–6.
38. Nelli F. Machine Learning with scikit-learn. In: Python data analytics. Springer; 2018. p. 313–347.
39. Nectoux P, Gouriveau R, Medjaher K, Ramasso E, Chebel-Morello B, Zerhouni N, et al. PRONOSTIA: An experimental platform for bearings accelerated degradation tests. In: IEEE international conference on prognostics and health management, PHM'12. IEEE Catalog Number: CPF12PHM-CDR; 2012. p. 1–8.
40. Mao W, He J, Zuo MJ. Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning. IEEE Trans Instrum Meas. 2019;69(4):1594–608.
41. Nguyen MH. Impacts of unbalanced test data on the evaluation of classification methods. ReCALL. 2019;100:90–00.
42. Shanbhag VV, Meyer TJ, Caspers LW, Schlanbusch R. Failure monitoring and predictive maintenance of hydraulic cylinder-state-of-the-art review. IEEE/ASME Trans Mechatron. 2021;26(6):3087–103.
43. Mota B, Faria P, Ramos C. Predictive maintenance for maintenance-effective manufacturing using machine learning approaches. In: 17th international conference on soft computing models in industrial and environmental applications (SOCO 2022) Salamanca, Spain, September 5–7, 2022, Proceedings. Springer; 2022. p. 13–22.
44. Bahador A, Du C, Ho CL, Jin Y, Dzulqarnain NA, Ng HP, et al. Condition monitoring for predictive maintenance of machines and processes in ARTC model factory. Implementing industry 40: the model factory as the key enabler for the future of manufacturing. 2021;p. 113–141.
45. Traini E, Bruno G, Lombardi F. Tool condition monitoring framework for predictive maintenance: a case study on milling process. Int J Prod Res. 2021;59(23):7179–93.
46. Lorena AC, Garcia LP, Lehmann J, Souto MC, Ho TK. How Complex is your classification problem? A survey on measuring classification complexity. ACM Comput Surv (CSUR). 2019;52(5):1–34.
47. Skrypnyk I. Irrelevant features, class separability, and complexity of classification problems. In: 2011 IEEE 23rd international conference on tools with artificial intelligence. IEEE; 2011. p. 998–1003.
48. Cano JR. Analysis of data complexity measures for classification. Expert Syst Appl. 2013;40(12):4820–31.
49. Liu Y, Hu Z, Zhang Y. Bearing feature extraction using multi-structure locally linear embedding. Neurocomputing. 2021;428:280–90.

Authors and Affiliations

Antonio Luca Alfeo[1,2] · Mario G. C. A. Cimino[1,2] · Guido Gagliardi[1,3,4]

✉ Guido Gagliardi
guido.gagliardi@phd.unipi.it

Antonio Luca Alfeo
luca.alfeo@unipi.it

Mario G. C. A. Cimino
mario.cimino@unipi.it

[1] Dept. of Information Engineering, University of Pisa, Pisa, Italy

[2] Research Center E. Piaggio, University of Pisa, Pisa, Italy

[3] Dept. of Information Engineering, University of Florence, Florence, Italy

[4] Dept. of Electrical Engineering, KU Leuven, Leuven, Belgium