electronics

Article

Anomaly Detection Methods for Industrial Applications: A Comparative Study

Maria Antonietta Panza *, Marco Pota and Massimo Esposito

National Research Council of Italy (CNR), Institute for High Performance Computing and Networking (ICAR), Via P. Castellino 111, 80131 Naples, Italy; marco.pota@icar.cnr.it (M.P.); massimo.esposito@icar.cnr.it (M.E.)
* Correspondence: mariaantonietta.panza@icar.cnr.it

Abstract: Anomaly detection (AD) algorithms can be instrumental in industrial scenarios to enhance the detection of potentially serious problems at a very early stage. Of course, the "Industry 4.0" revolution is fostering the implementation of intelligent data-driven decisions in industry based on increasingly efficient machine learning (ML) algorithms. Most well-known AD methods use a supervised learning approach focusing on fault classification. They assume the availability of labeled data for both normal and anomalous classes. However, in many industrial environments, a labeled set of anomalous data instances is more challenging to obtain than a labeled set of normal data. Hence, this work implements an unsupervised approach based on two different methods using a typical benchmark bearing-fault dataset. The first method relies on the manual extraction of typical vibration metrics provided as input to an ML algorithm. The second one is based on a deep learning (DL) approach, automatically learning latent representation from raw data. The performance metrics demonstrate that both approaches can distinguish the state of a bearing from normal to faulty. DL methodology proves a higher accuracy rate in recognizing faults and a better ability to provide information about the fault size.

Keywords: anomaly detection; OC-SVM; autoencoders; bearing fault

check for updates

Citation: Panza, M.A.; Pota, M.; Esposito, M. Anomaly Detection Methods for Industrial Applications: A Comparative Study. *Electronics* **2023**, *12*, 3971. https://doi.org/10.3390/electronics12183971

Academic Editor: Ahmed Abu-Siada

Received: 11 August 2023
Revised: 11 September 2023
Accepted: 17 September 2023
Published: 20 September 2023

1. Introduction

The detection of anomalies is gaining importance in industrial scenarios, driven by the relatively cheap solutions the "Industry 4.0" revolution has made available for data acquisition, storage, cloud computing, and efficient machine learning algorithms. Hence, the growing interest in implementing adequate data-driven decisions is pulling the recent development of several intelligent computing methods for anomaly detection (AD) in industry [1].

Industrial facilities deteriorate over time due to usage and normal wear. Such deterioration must be identified early to prevent possible damages and losses and to maximize machine productivity and remaining useful life (RUL) while reducing maintenance and repair costs. Machinery is then usually instrumented with sensors dedicated to monitoring the performance of industrial components. The data collected by these sensors may include temperature, pressure, vibration, acoustic emission, and motor current [2]. Industrial damage detection typically deals with defects in mechanical components such as motors, turbines, pipeline oil flow, and physical structures.

Most existing machine condition monitoring techniques involve sample/signal analysis and use an indicator to detect fault by comparing it to a predefined alarm threshold. This approach requires expertise in data analysis and knowledge of the machine and its operation. Moreover, the threshold-based technique does not ensure the detection of all possible faults, since some of them involve complex behaviors, such as unwanted oscillations or abnormal increasing/decreasing speed of monitored signals, which remain within thresholds. Recent progress in Artificial Intelligence (AI) technologies has brought several advantages by developing automated fault detection procedures based on available data.

The lack of reliable tagged data may represent a typical issue in many industrial environments, where a labeled set of anomalous data instances covering all possible types of anomalous behavior is usually more challenging to obtain than a labeled set of normal data. In addition, the anomalous behavior is often dynamic (new types of anomalies might arise) and sometimes would result in a catastrophic event, e.g., [3]. For these reasons, semi-supervised and unsupervised learning approaches are the most widely applicable in the industrial domain.

Hence, in this work, an unsupervised approach for AD is investigated and applied to a typical fault detection problem in the industrial field, i.e., the bearings diagnosis. As some of the most relevant rotating machinery components, rolling element bearings play an essential role in adequately functioning various industrial facilities. Bearings are the best location for measuring machinery vibration, where the machine's primary dynamic loads and forces are applied. Therefore, condition monitoring and fault diagnosis of this critical component can represent the machine's condition.

The methodology is applied to the Case Western Reserve University (CWRU) benchmark dataset, widely used for developing bearing fault detection approaches. The dataset provides vibration data collected for normal and faulty bearings. Most approaches applied to the CWRU bearing dataset are based on supervised learning and aim to classify different bearing states. This work aims to outline a methodology based on unsupervised learning to monitor online bearing health state and detect possible anomalous behaviors for undertaking preventive countermeasures as soon as the anomaly occurs.

Two different approaches are analyzed for this scope, the first based on the manual extraction of typical vibration metrics to be provided as input to a machine learning (ML) algorithm, the second based on a deep learning (DL) method able to extract useful features from raw data automatically. The two approaches are compared by testing a representative method for each method class, evaluating their effectiveness and efficiency and their strengths and weaknesses in an industrial context.

The article is organized as follows. Section 2 provides a brief background analysis and a literature review of the standard anomaly detection methods applied to the industrial domain. In Section 3, the dataset and the applied methods are described in detail, whereas in Section 4 the obtained results are presented. In Section 5, conclusions are drawn from this study by outlining further research developments.

2. Background and Related Work

As a rule, AD methodologies aim to identify data patterns that differ significantly from expected normal behavior [4]. These nonconforming patterns are most referred to as anomalies or outliers, sometimes interchangeably. Several key aspects need to be considered in developing an adequate AD methodology.

The nature of input data represents a primary factor for an AD technique. Input is generally a collection of data instances. Each data instance can be described using a set of attributes (also referred to as variable, characteristic, feature, or dimension) and might consist of only one attribute (univariate data) or multiple attributes (multivariate data). The attributes can be of different types, such as binary, categorical, or continuous.

Another essential aspect of AD problems regards the type of anomaly. Anomalies can be classified into point, contextual, and collective anomalies [5]. An individual data instance is called a point anomaly if it can be considered anomalous concerning the rest of the data. That is the simplest type of anomaly and the most popular in anomaly detection research. A contextual anomaly occurs if a data instance is anomalous in a specific context but not otherwise. Then, data instances' contextual and behavioral attributes must be specified as part of the problem formulation. A collective anomaly is defined as a collection of anomalous related data instances for the entire dataset. In this latter case, the individual data instances in the collection may not be anomalies alone, but their occurrence together as a collection is anomalous.

The label of a data instance determines whether that instance can be classified as normal or anomalous. The availability of labeled data for training/validation of models used by AD techniques is usually a significant issue. A human expert often does manual labeling, and obtaining an accurate and well-representative labeled training dataset can be prohibitively expensive. Hence, based on the availability or unavailability of labeled data instances, AD techniques operate in supervised, semi-supervised, and unsupervised modes. Supervised techniques assume the availability of a training dataset that has labeled instances for normal and anomaly classes. A typical approach in such cases is to build a predictive model for normal vs. anomaly classes and use the model to determine to which class any unseen data instance belongs. Unsupervised techniques assume that the training data wholly consist of normal samples, i.e., samples belonging to the class chosen as normal. On the other hand, semi-supervised AD refers to experiments where, in addition to the training set samples belonging only to the normal class, a few labeled anomalies are available and considered by the training objective to help improve the detection [6].

Another key factor for any AD technique is how to report the anomalies. Typically, the outputs may be scores or labels. Scoring methods allow defining a so-called Anomaly Score (AS), a quantitative index assigned to each instance in the test data depending on the observation's degree of 'outlierness'. Thus, the output of such methods is a ranked list of anomalies. An analyst may choose to analyze the top few anomalies or use a cutoff threshold to select the most relevant ones. In industrial applications, the AS can be exploited as a health factor for real-time system monitoring. Inspections or other maintenance actions can be undertaken when the AS overcomes a pre-determined threshold. Other methods directly assign a label (normal or anomalous) to each test instance.

AD uses many approaches, including model-based and data-driven methods. However, the complexity of present-day industrial systems and the rising cost of model building lead to limitations on using model-based techniques. Recently, data-driven methods have gained attention because of their ability to derive models without prior knowledge of the application domain, also fostered by recent advances in sensor technologies, data processing, and computation power growth.

Several data-driven algorithms have been applied to AD tasks [7]. Unsupervised rather than supervised methods are often adopted in industrial applications due to the inability, in many realistic cases, to collect a representative set of defective data corresponding to several faulty operations.

Classical ML approaches have been successfully adopted for unsupervised AD. In particular, One-Class Support-Vector Machines (OC-SVMs) are one of the most robust unsupervised outlier detection algorithms and have been successfully used for fault diagnosis by the ML community over the past 15 years. That is due to its excellent generalization ability and high accuracy using a relatively small number of samples [8].

In [9], an automatic method for bearing fault detection and diagnosis is presented based on a one-class SVM trained using only historical data under normal conditions.

The work in [10] proposes novel data-driven architectures trained exclusively on healthy signals, combining LSTM regressors and OC-SVM classifiers, to develop an automated algorithm to identify any abnormal mechanical behavior captured by vibration measurements.

The scientific community has adopted unsupervised DL methods since they outperform conventional ML techniques, which are limited in processing raw data by the need for careful engineering and considerable domain expertise for transforming the data into a suitable internal representation or feature vector. Conversely, DL methods are representation learning methods that allow a machine to be fed with raw data and to automatically discover the representations needed for detection or classification [11]. Several DL approaches in unsupervised AD tasks exploit the autoencoders (AEs) framework. The study in [12] proposes an intelligent fault diagnosis approach that uses a deep neural network (DNN) based on a stacked denoising autoencoder applied to unlabeled data. Authors in [13] propose an unsupervised method for diagnosing faults of electric motors by using a novelty detection approach based on deep AEs. In [14], a comparison of different AE architectures is presented, including vanilla and variational AEs with different types and number of layers, for AD of a real manufacturing industrial furnace.

An innovative DL-based model using AEs is developed in [15], which improves anomaly detection accuracy in bearing condition monitoring.

AD finds extensive use in various application domains, especially industrial damage detection. Among all the publicly available datasets, CWRU Bearing Data Center data have become a standard reference for testing new diagnostic bearing algorithms [16,17]. In most of the research papers that use the CWRU bearing dataset, the aim is to apply unsupervised learning approaches to classify the fault category. Moreover, in the literature, the main concern is the diagnosis of data records. Although classic bearing fault features dominate some signals, others are unclear or display other fault symptoms. The study in [18] provides a benchmark analysis applying three diagnostic methods that use the squared envelope spectrum. The authors found that the data records ranged from easily diagnosable to undiagnosable with applied methods. The ball faults are among the most difficult to diagnose. The same outcome is verified in [19], where envelope analysis is applied to the signal filtered in the frequency band with the most diagnostic information.

In the current research, both OC-SVM and AEs-based approaches are then implemented, assessing and comparing their performance for unsupervised fault detection in the CWRU bearing dataset.

3. Materials and Methods

3.1. Dataset

In order to test the performance of the algorithms used in this study, the CWRU dataset provided by the Case Western Reserve University Bearing Data Center [20] is singled out among all the reference datasets available in the public domain.

The CWRU dataset is a standard benchmark in the context of fault detection. It provides ball-bearing test data for normal and faulty bearings. As shown in Figure 1, the layout of the test rig consists of a 2 hp Reliance Electric motor driving a shaft on which a torque transducer and an encoder are mounted. The test bearings support the motor shaft, and torque is transmitted to the shaft via a dynamometer and an electronic control system.

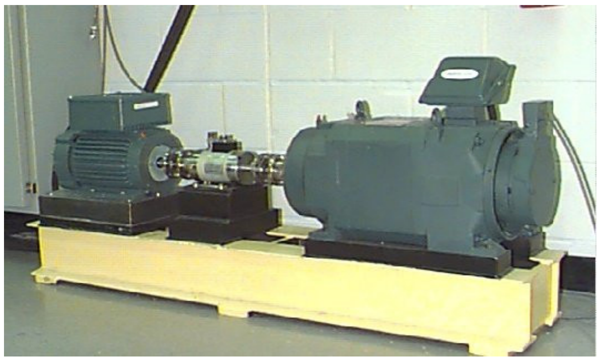


Figure 1. CWRU bearing test rig.

During the experiments, acceleration was measured in the vertical direction on the housing of the drive-end bearing. Some tests also measured it on the fan-end bearing housing and the motor supporting base plate. Faults deep 0.011 inch and ranging in diameter from 0.007 to 0.028 inch (0.18–0.71 mm) were seeded on the motor's bearings using electro-discharge machining (EDM). The defects were introduced one at a time on the rolling elements and the inner and outer raceway, and each faulty bearing was reinstalled on the test rig. Vibration data were recorded for motor loads of 0 to 3 hp (motor speeds of 1797 to 1720 rpm) at the sampling frequency of 12 kHz and 48 kHz, depending on the test condition.

For further test setup details, please refer to the CWRU Bearing Data Center website [20].

This study uses drive-end acceleration data collected at the sample rate of 12 kHz in normal baseline conditions and when inner race, outer race, and ball defects are introduced on the fan-end bearing. For the outer race defect, we consider only data related to the 'orthogonal' (3.00 o'clock) fault position relative to the load zone. It is worth noticing that, as reported in [18], even though the only load borne by the bearings (in theory) is the static gravitational load from the weight of the shaft and any attached components, there is evidence that there could be a dynamic load superimposed on the static load. The dataset names and details are shown in detail in Table 1.

Table 1. Dataset names and parameters.

Reference Name	Fault Diameter, in	Motor Load, hp	Approx. Motor Speed, rpm	Data Points
Normal_1	-	1	1772	480,000
Fault_B007_1/Fault_B014_1/Fault_B021_1	0.007/0.014/0.021	1	1772	120,000
Fault_IR007_1/Fault_IR014_1/Fault_IR021_1	0.007/0.014/0.021	1	1772	120,000
Fault_OR007_1/Fault_OR014_1/Fault_OR021_1	0.007/0.014/0.021	1	1772	120,000
Normal_2	-	2	1750	480,000
Fault_B007_2/Fault_B014_2/Fault_B021_2	0.007/0.014/0.021	2	1750	120,000
Fault_IR007_2/Fault_IR014_2/Fault_IR021_2	0.007/0.014/0.021	2	1750	120,000
Fault_OR007_2/Fault_OR014_2/Fault_OR021_2	0.007/0.014/0.021	2	1750	120,000
Normal_3	-	3	1730	480,000
Fault_B007_3/Fault_B014_3/Fault_B021_3	0.007/0.014/0.021	3	1730	120,000
Fault_IR007_3/Fault_IR014_3/Fault_IR021_3	0.007/0.014/0.021	3	1730	120,000
Fault_OR007_3/Fault_OR014_3/Fault_OR021_3	0.007/0.014/0.021	3	1730	120,000

Differences in the records can be noted in Figure 2, which shows the raw time signals for the bearing normal condition and each fault category at 3 hp motor load.

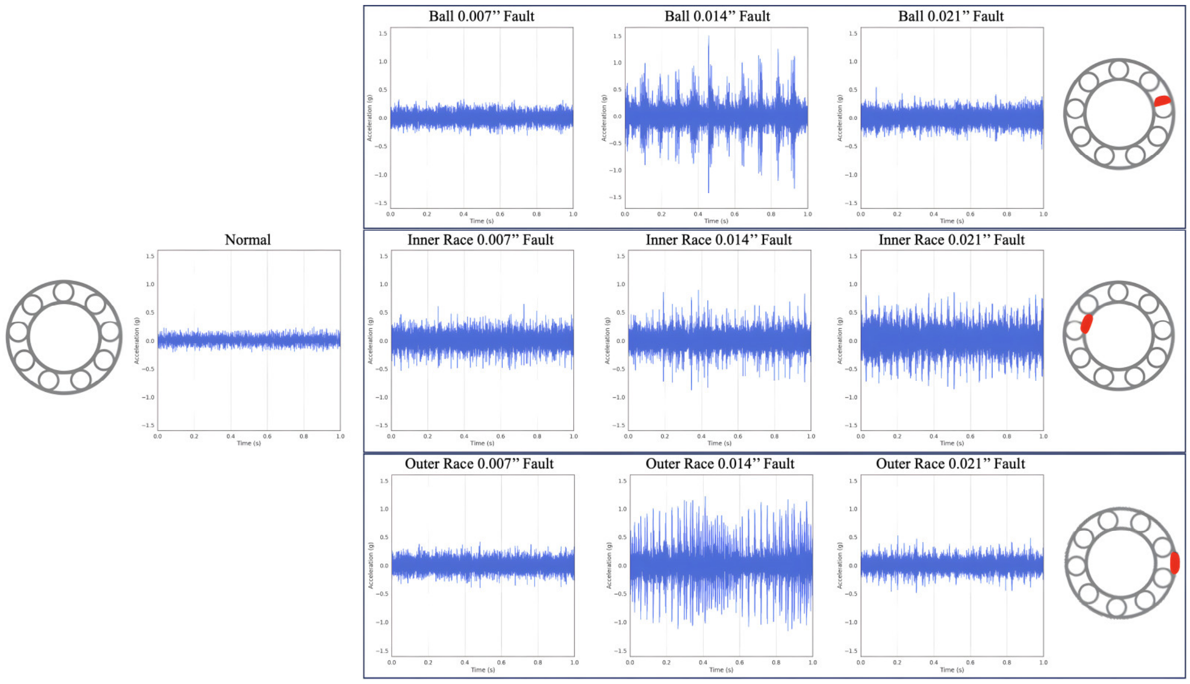


Figure 2. Raw vibration data in bearing normal baseline and faulty states at 3-hp motor load.

3.2. Methodology

The paper presents an exploratory analysis of different methods' ability to solve a typical anomaly detection problem. For a better understanding, Figure 3 shows the flow chart of the methodology according to the two applied approaches.

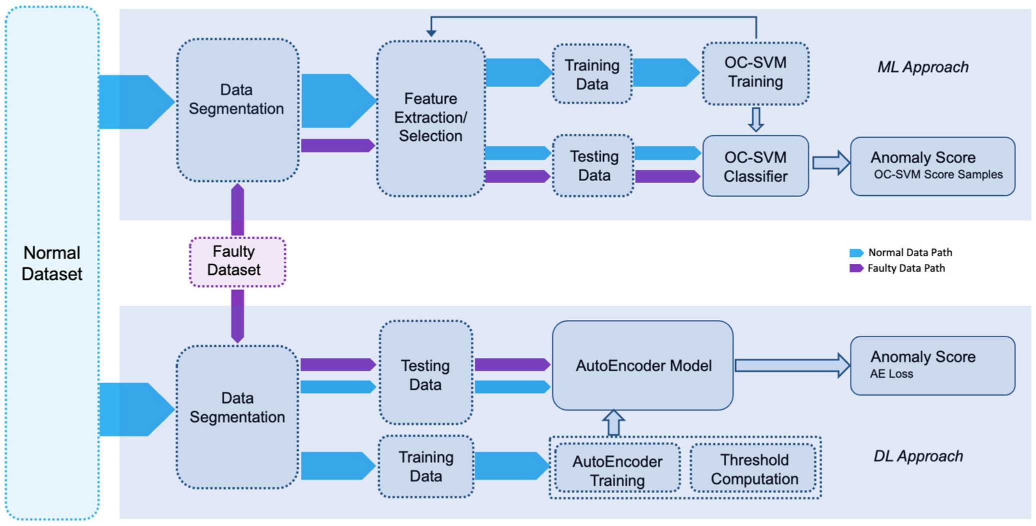


Figure 3. Methodology flow chart.

The diagram highlights that the techniques operate in unsupervised mode, as the training data entirely consist of samples belonging to the class labeled as normal. In contrast, anomaly classes are used in testing the methods.

The following sections describe the typical data segmentation phase and the two alternative ML and DL approaches.

3.2.1. Data Segmentation

Measurements are first segmented to build windows with a fixed time length. The selection of the window length is an essential issue since it must be set to guarantee, on the one hand, enough data samples for frequency data processing and, on the other hand, a small step size to ensure online monitoring of bearing health state.

The window value is then chosen according to the motor speed and the sampling rate. The motor speed ranges from 1730 to 1772 rpm, meaning approximately 30 rotations/s. As the sampling frequency is set to 12,000 samples/s, about 400 data points are collected in one revolution of the bearing.

For ensuring that the length of a single sample can completely and accurately reflect vibration distribution in the frequency domain with a proper resolution, setting a window length of 800 data points seems appropriate. Given the beneficial effects of data augmentation through overlapping [21], 25% overlap is used, meaning that the stride length includes 600 data points. Data are then segmented into 800 windows for each normal class and 200 windows for each anomaly class.

Figure 4 shows the first overlapping sequences for a typical dataset time series.

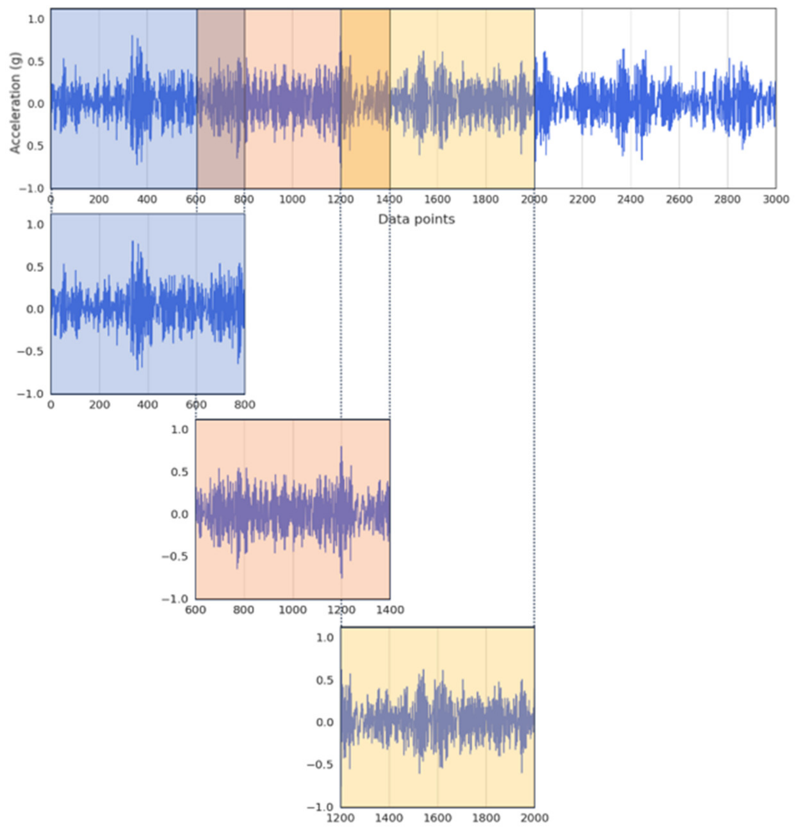


Figure 4. Sliding window for data segmentation.

3.2.2. Machine Learning Approach

The ML methodology implemented in this study is based on extracting different features from the vibration time series and their use for anomaly detection. Specifically, the features extracted from the raw measurements corresponding to normal operation are used to train an OC-SVM model for AD. The trained model is then tested on the features extracted from new time series data corresponding to normal and anomalous operations.

In more detail, the learning stage allows the nominal vibration behavior model to be constructed. The processing phases can be synthesized as follows:

- A pre-processing converts the time-series acquisitions into a multidimensional scatter plot, each point being a vector of features and representing a time window;
- The dimension of the scatter plot is reduced through Pearson correlation;
- A decision frontier is fitted around the scatter plot by an OC-SVM algorithm.

Once the nominal behavior model is defined, AD is performed on new time series data by applying the same pre-processing and dimension reduction and determining whether the resulting data lie inside or outside the decision frontier.

A custom code is written in Python 3.10.12, using the scikit-learn library (version 1.2.2) [22] to build and train the ML algorithm.

Feature Engineering

Feature engineering uses knowledge about the data to improve the algorithm by applying transformations to the data before providing it as input to the model. The basic idea is making a problem easier by expressing it more simply. A deep understanding of the problem is thus needed [23].

Feature engineering is a general concept that spans different topics, including feature transformation, feature generation, feature selection, feature analysis, and evaluation [24]. Feature transformation refers to constructing new features from existing ones and often uses mathematical mapping. Feature generation, sometimes referred to as feature extraction, is usually about generating new features in addition to the original ones. The final goal is to extract useful information from the raw signals. Unfortunately, not all features are meaningful and informative about machine conditions. So, feature selection is about selecting a small subset of features from a more extensive set. That is very important because providing irrelevant and or highly correlated features as input data may spoil the algorithm's generalization ability, resulting in overfitting [25]. Furthermore, the reduced dimensionality of the feature set makes the use of specific algorithms computationally feasible and improves their performance. Feature selection may also foster the algorithm interpretability. Feature analysis and evaluation refer to concepts, methods, and measures for evaluating the usefulness of features and feature sets and often are included in the feature selection process. All machine learning tasks, including AD, can use practical feature engineering.

In time series AD, a mainstream approach is extracting multi-domain features through signal processing. The commonly used features in the literature can be extracted from time-domain, frequency-domain, and time-frequency analysis [26–28].

Traditional time-domain analysis calculates characteristic features from time series signals as descriptive statistics such as mean, peak, peak-to-peak interval, standard deviation, crest factor, and high-order statistics such as root mean square, skewness, and kurtosis. These features are usually called time-domain features. The calculation formulas of the most common ones used in this study are shown in Table 2.

Table 2. Time-domain features.

Feature	Formula	Feature	Formula
Max.	$x_{max} = \max_i(x_i)$	RMS	$x_{RMS} = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(x_i)^2}$
Min.	$x_{min} = \min_i(x_i)$	Crest factor	$x_{CF} = \frac{x_p}{x_{RMS}}$
Mean	$x_{mean} = \frac{1}{N_s}\sum_{i=1}^{N_s}x_i$	Shape factor	$x_{SF} = \frac{x_{RMS}}{\frac{1}{N_s}\sum_{i=1}^{N_s}\lvert x_i\rvert}$
Std.	$x_{std} = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(x_i - x_{mean})}$	Skewness	$x_{skew} = \frac{\frac{1}{N_s}\sum_{i=1}^{N_s}(x_i - x_{mean})^3}{\left[\frac{1}{N_s}\sum_{i=1}^{N_s}(x_i - x_{mean})^2\right]^{3/2}}$
Peak	$x_p = \max_i(\lvert x_i\rvert)$	Kurtosis	$x_{kurt} = \frac{\frac{1}{N_s}\sum_{i=1}^{N_s}(x_i - x_{mean})^4}{\left[\frac{1}{N_s}\sum_{i=1}^{N_s}(x_i - x_{mean})^2\right]^2}$
Peak-to-peak	$x_{p2p} = x_{max} - x_{min}$	Impulse factor	$x_{IF} = \frac{x_p}{\frac{1}{N_s}\sum_{i=1}^{N_s}\lvert x_i\rvert}$

Frequency analysis is the most used method for analyzing a vibration signal utilizing its frequency spectrum [29]. The most basic type of frequency analysis is an FFT, or Fast Fourier Transform, which converts a signal from the time domain into the frequency domain. In a complex signal, the FFT helps engineers determine the frequencies being excited and their amplitudes, highlighting the changes in frequency and amplitude and the harmonic excitation in a defined frequency range. In order to view the energy distribution across the frequency spectrum, the PSD, or Power Spectral Density, must be calculated from the FFT. The PSD is the measure of signal's power content over frequency and the most common tool for analyzing broadband random vibration. PSD and FFT are tools for measuring and analyzing a signal's frequency content. However, whereas the FFT transfers time data to the frequency domain, highlighting changes in frequency values, the PSD takes another step and calculates the power, or strength, of the frequency content. The magnitude of the PSD is normalized to the frequency bin width. Therefore, the resulting value is independent of the bandwidth and suitable for analyzing and comparing random vibration signals, whose frequency spectra can be overlaid and compared independently of the spectral resolution. Among all the commonly used frequency-domain features, in this study, the following features, calculated from the PSD frequency spectrum, are considered:

- The peak frequency;
- The peak amplitude;
- The RMS level in the low-frequency range (below 65 Hz);
- The RMS level in the mid-frequency range (between 65 Hz and 300 Hz);
- The RMS level in the high-frequency range (above 300 Hz);
- The overall RMS level.

The calculation formulas of the frequency domain features are reported in Table 3, where s(k) is the PSD spectrum for k = 1, 2, . . ., K, K is the number of spectrum lines, and f_k is the frequency value of the kth spectrum line.

Table 3. Frequency-domain features.

Feature	Formula
Peak frequency	$f_{peak} = f_k : \max_k(s_k)$
Peak amplitude	$s_{peak} = \max_k(s_k)$
$RMS_{f\,<\,65\ Hz}$	$s_{(f\,<\,65\ Hz)} = \sqrt{\frac{1}{k_{65}}\sum_{k=1}^{k_{65}}(s_k)^2}$
$RMS_{65\ Hz\,<\,f\,<\,300\ Hz}$	$s_{(65\ Hz\,<\,f\,<\,300\ Hz)} = \sqrt{\frac{1}{k_{300}-k_{65}}\sum_{k=k_{65}}^{k_{300}}(s_k)^2}$
$RMS_{f\,>\,300\ Hz}$	$s_{(f\,>\,300\ Hz)} = \sqrt{\frac{1}{K-k_{300}}\sum_{k=k_{300}}^{K}(s_k)^2}$
$RMS_{overall}$	$s_f = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(s_k)^2}$

Time-frequency analysis allows obtaining features usually extracted by wavelet transform (WT) or empirical mode decomposition (EMD), which can reflect the health status of machines under non-stationary operating conditions.

Since the CWRU bearing-test data do not include any transient condition, only time-domain and frequency-domain features are extracted and fed as input into the ML algorithm. Before that, the most discriminative features must be selected from the feature set, and redundant features must be discarded to improve the model performance and avoid the curse of dimensionality. Here, the Pearson correlation matrix is computed. It measures the strength and direction of the relationship between two variables and thus allows the identification of the correlated features. The screening process analyzes the correlation coefficients of all the couples of features. For each couple, one of the features is discarded if they are highly correlated, and the model performance is not penalized. The least redundant features are chosen, reducing the total number of features from 18 to 10. The ten chosen features are RMS, max, peak-to-peak, skewness, kurtosis, and shape factor as time-domain features, peak amplitude, RMS level in the mid-frequency range, RMS level in the high-frequency range, and overall RMS level as frequency domain features.

OC-SVM Algorithm

Support Vector Machine (SVM) is one of the most successful machine learning techniques that can be used in various classification applications. The technique classifies data by finding the hypersurface that best separates the features into different domains. If the training data are not linear, it uses a kernel function to map it into a higher-dimension space. The instances closest to the separating hyperplane are called support vectors. In contrast to traditional SVM, OC-SVM is an unsupervised method that does not require target labels for the training process, as it aims to determine the behavior of data from a single known class, the target class, and tag the elements that lie on the other side of the decision boundary as outliers. For more detailed information, refer to Schölkopf et al. [30,31].

This study implements an OC-SVM classifier on the selected feature set for each motor load condition. A radial basis function (RBF) is used as the kernel function because of its ability to model complex and non-linear decision boundaries [32]. The RBF kernel measures the similarity between two data points as a function of the Euclidean distance between them. The kernel function is defined as:

$$K(x, y) = \exp(-\gamma\ ||x - y||^2), \tag{1}$$

where x and y are two data points, γ is the kernel coefficient that determines the width of the kernel function, and $||x - y||$ is the Euclidean distance between x and y. Choosing the correct value of γ is crucial for achieving good performance with the RBF kernel. γ defines the range of influence of a single training point. It can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. A more considerable γ value makes the kernel function more peaked, leading to a more complex decision boundary that better captures the details of the training data; conversely, a more minor γ value results in a smoother decision boundary that may be more generalizable to new data. In addition to γ, the RBF kernel has another hyperparameter called ν. This regularization parameter controls the trade-off between achieving a good fit to the training data and a simple decision boundary. Larger values of ν allow for more complex decision boundaries, which can lead to overfitting, whereas smaller values of ν may result in underfitting. Therefore, both parameters control complexity in some way.

In this case, the default value of γ in scikit-learn is selected as the kernel coefficient (equal to the inverse of the product of the number of features and the variance of the data) since the dataset is not scaled to have a unit standard deviation. Because only the normal class needs to be modeled, the value of ν is tuned to the approximate ratio of outliers in the data. In particular, coherently with [30], $\nu = 0.01$ is chosen to be consistent with the estimated fraction of abnormal instances. The AS, by definition, is computed through the scoring function of the samples.

In order to differentiate the classifier depending on the specific anomaly class to be detected, several OC-SVM models are trained in parallel by changing the input features while keeping the model tuning parameters fixed. That led to identifying, for each fault category, the set of features shown in Table 4, ensuring the best model performance.

Table 4. Reduced set of features for each anomaly class.

Anomaly Class	**Features**
Ball Fault	x_{RMS}, x_{p2p}, x_{skew}, $s_{(65Hz<f<300Hz)}$, $s_{(f>300Hz)}$
Inner Race Fault	x_{max}, x_{RMS}, x_{p2p}, x_{skew}, x_{SF}, s_{peak}, $s_{(f>300\ Hz)}$
Outer Race Fault	x_{max}, x_{RMS}, x_{p2p}, x_{kurt}, $s_{(f>300\ Hz)}$, s_f

The feature set is split into training and testing sets, as reported in Table 5. The algorithm is trained using only normal data samples and tested using both normal and anomalous data samples. Input consists of multivariate data, i.e., the multiple features calculated from the acceleration time series at the drive end of the motor housing.

Table 5. Data splitting for OC-SVM algorithm.

Set	Normal Samples	Anomalous Samples
Training	600	0
Test	200	200

3.2.3. Deep Learning Approach

DL methods have become increasingly popular in recent years because of their success in many domains, such as image classification, speech recognition, and natural language processing. The great success of DL mainly comes from specially designed structures of deep nets, which can learn discriminative non-linear features. Hence, most existing DL methods can be used as powerful feature learning/extraction tools, as the latent features extracted by DL algorithms are newly learned representations.

DL approach is also used in AD problems to detect time series anomalies. Among the different methods for time series data in the literature, the reconstruction methods [33] build a model of normal behavior by encoding subsequences of a normal training time series in a (low dimensional) latent space. In order to detect anomalies in a test time series, subsequences of the test series are reconstructed from the latent space, and their values are then compared to the original series values. Since the model is trained only on normal data, the model cannot reconstruct anomalous subsequences in the test series. Therefore, the difference between the original and the reconstructed subsequences can provide an anomaly score.

A representative method of this category used for the present evaluation is the autoencoder (AE) model. An AE is a neural network trained to learn latent representation able to reconstruct its input [34]. Generally, an AE comprises two parts, i.e., an encoder f(·) and a decoder g(·). An illustration of AE is shown in Figure 5.

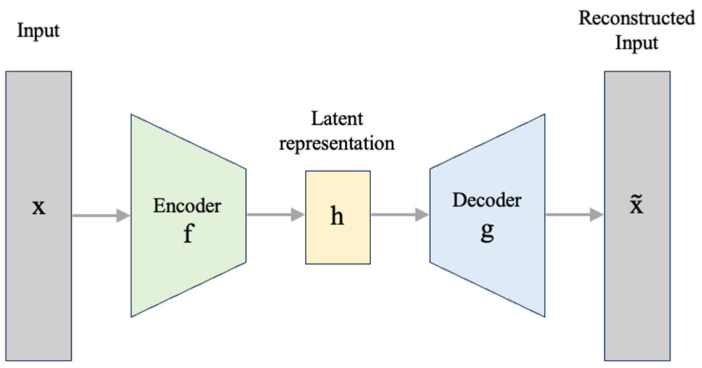


Figure 5. AE framework.

The encoder maps the input $x \in R^m$ to latent representation $h \in R^d$ as $h = f(x)$ and $f(\cdot)$ is usually a one-layer neural network, i.e., $f(x) = s(Wx + b)$, where $W \in R^{d\,x\,m}$ and $b \in R^d$ are the weights and bias of the encoder. $s(\cdot)$ is a non-linear function such as sigmoid and tanh. A decoder maps back the latent representation h into a reconstruction $\tilde{x} \in R^m$ as $\tilde{x} = g(h)$ and $g(\cdot)$ is given as $g(h) = s(W'h + b')$, where $W' \in R^{m\,x\,d}$ and $b' \in R^m$ are the weights and bias of the decoder. Note that the prime symbol does not indicate matrix transposition. The parameters of the autoencoder $\theta = \{W, b, W', b'\}$ are optimized to minimize the reconstruction error. Depending on the distribution assumptions of the input, the reconstruction error can be measured in different ways. The most widely used reconstruction error is the squared error $\mathcal{L}\left(x, \tilde{x}\right) = \left\|x - \tilde{x}\right\|_2^2$.

The autoencoder can be used to extract valuable features by forcing the latent representation h to have a smaller dimension than x, i.e., $d < m$. Learning such a representation forces the autoencoder to capture the most salient features of the training data.

An AE algorithm is trained over the normal vibration data for each considered motor load condition to build a model of bearing nominal behavior. The algorithm is coded in Python 3.10.12, using the Keras computational library (version 2.12.0) [35] to implement the AE architecture.

After the segmentation process described in Section 3.2.1, the dataset is split into training and testing sets. The model is trained using only normal data samples and tested over normal and anomalous data samples. Table 6 reports the data splitting information. Now, the input consists of univariate data, directly the vibration time series collected by the accelerometer at the drive end of the motor housing.

Table 6. Data splitting for AE algorithm.

Set	Normal Samples	Anomalous Samples	Size
Training	600	0	800
Test	200	200	800

In the AE architecture, input and output vectors have dimension T = 800, whereas the latent space has dimension D = 24. The encoder and the decoder are constituted by six fully connected layers, consisting of a complete feed-forward connection without activation function, as detailed in Table 7.

Table 7. The fully connected AE structure.

Component	Architecture
Encoder $x_{800} \rightarrow h_{24}$	$\rightarrow FC_{800} \rightarrow FC_{400} \rightarrow FC_{200} \rightarrow FC_{100} \rightarrow FC_{50} \rightarrow FC_{24} \rightarrow$
Decoder $h_{24} \rightarrow \tilde{x}_{800}$	$\rightarrow FC_{24} \rightarrow FC_{50} \rightarrow FC_{100} \rightarrow FC_{200} \rightarrow FC_{400} \rightarrow FC_{800} \rightarrow$

After training, a limit threshold to the reconstruction error must be adequately set to evaluate the algorithm's performance.

One of the simplest threshold selection methods uses the three-sigma rule of thumb [36], expressing a conventional heuristic. Almost all values will lie within three standard deviations of the mean for normally distributed data. Thus, the 99.7% probability can be treated as near certainty. Therefore, data points that fall outside three-sigma limits can be considered outliers. Figure 6 shows the threshold definition in this work, set to three standard deviations of the mean of the loss computed over training normal samples.

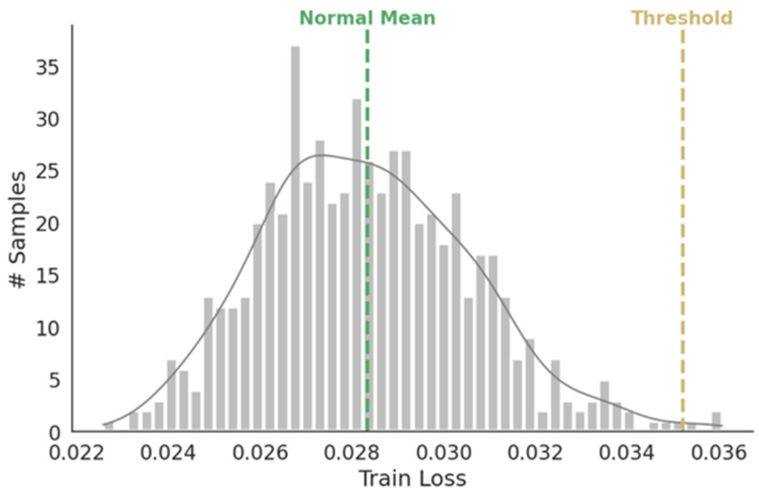


Figure 6. AE threshold definition.

4. Results and Discussion

First, the models' performances are assessed using some of the most used evaluation metrics listed in the following.

- Accuracy: the fraction of correctly predicted instances over the total number of instances.

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{2}$$

- Precision: the fraction of positive instances the model correctly predicts, useful when the cost of false positives is high.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

- Recall: the fraction of positive instances correctly predicted by the model out of all the positive instances in the data, useful when the cost of false negatives is high.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

- F1 Score: the harmonic mean of precision and recall, a balanced measure considering false positives and false negatives.

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{5}$$

- AUC-ROC: the "Area Under the Receiver Operating Characteristic" curve, i.e., the probability that a randomly selected positive instance will be ranked higher than a randomly selected negative instance [37].

Tables 8–10 show the values of the evaluation metrics across the two different approaches for each anomaly class and load condition. Test performances are computed on five different random train-test splits; therefore, metrics values are here expressed in terms of mean and standard deviation. The results reveal that, on average, both methods perform well. In almost all cases, OC-SVM exhibits the best performance in detecting bearing inner race and outer race defects.

At the same time, the AE reconstruction method performs better for the diagnosis of ball fault cases. For this type of anomaly, the ML approach shows lower accuracy and recall values when the fault diameter is 0.007 inch. It is worth noticing that, unlike OC-SVM, the approach based on AEs allows 100% recall (FN = 0) for all experiments, presenting a 100% rate in recognizing faults.

For better comprehension, the simulation results of compared methods are also analyzed in terms of AS and confusion matrices, which help to understand the types of errors the model makes, showing the number of true positives, true negatives, false positives, and false negatives. Specifically, the OC-SVM score samples and the AE loss are used as anomaly scores, and their values are plotted for 200 normal and 200 faulty test samples. Some of the most representative results are shown in Figures 7–9, where each black dot corresponds to a normal sample. In contrast, a blue, green, and red one corresponds to an abnormality, and yellow diamond points, if present, mark false positives and false negatives.

Table 8. OC-SVM and AE performance metrics in detecting inner race defects.

Inner Race Faults (IR007/IR014/IR021)						
Method	**Motor Load, hp**	**Performance Metrics**				
		Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
OC-SVM	1	0.993/0.996/0.996 ±0.004/±0.004/±0.004	0.991/0.991/0.991 ±0.008/±0.008/±0.008	0.995/1/1 0/0/0	0.993/0.995/0.995 ±0.004/±0.004/±0.004	0.993/0.996/0.996 ±0.004/±0.004/±0.004
	2	0.995/0.997/0.997 ±0.007/±0.007/±0.007	0.994/0.994/0.994 ±0.013/±0.013/±0.013	0.995/1/1 0/0/0	0.994/0.997/0.997 ±0.007/±0.007/±0.007	0.995/0.997/0.997 ±0.007/±0.007/±0.007
	3	0.996/0.996/0.996 ±0.003/±0.003/±0.003	0.991/0.991/0.991 ±0.007/±0.007/±0.007	1/1/1 0/0/0	0.995/0.995/0.995 ±0.003/±0.003/±0.003	0.996/0.996/0.996 ±0.003/±0.003/±0.003
AE	1	0.984/0.984/0.984 ±0.003/±0.003/±0.003	0.969/0.969/0.969 ±0.006/±0.006/±0.006	1/1/1 0/0/0	0.984/0.984/0.984 ±0.003/±0.003/±0.003	0.984/0.984/0.984 ±0.003/±0.003/±0.003
	2	0.987/0.987/0.987 ±0.003/±0.003/±0.003	0.975/0.975/0.975 ±0.006/±0.006/±0.006	1/1/1 0/0/0	0.987/0.987/0.987 ±0.003/±0.003/±0.003	0.987/0.987/0.987 ±0.003/±0.003/±0.003
	3	0.981/0.981/0.981 ±0.010/±0.010/±0.010	0.962/0.962/0.962 ±0.018/±0.018/±0.018	1/1/1 0/0/0	0.981/0.981/0.981 ±0.009/±0.009/±0.009	0.981/0.981/0.981 ±0.009/±0.009/±0.009

Table 9. OC-SVM and AE performance metrics in detecting outer race defects.

Outer Race Faults (OR007/OR014/OR021)						
Method	**Motor Load, hp**	**Performance Metrics**				
		Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
OC-SVM	1	0.990/0.990/0.990 ±0.007/±0.007/±0.007	0.979/0.979/0.979 ±0.014/±0.014/±0.014	1/1/1 0/0/0	0.989/0.989/0.989 ±0.007/±0.007/±0.007	0.990/0.990/0.990 ±0.007/±0.007/±0.007
	2	0.991/0.991/0.985 ±0.007/0.007/0.004	0.982/0.982/0.982 ±0.014/±0.014/±0.014	1/1/0.987 0/0/±0.006	0.991/0.991/0.984 ±0.007/±0.007/±0.005	0.991/0.991/0.985 ±0.007/±0.007/±0.004
	3	0.990/0.990/0.990 ±0.003/±0.003/±0.003	0.980/0.980/0.980 ±0.006/±0.006/±0.006	1/1/1 0/0/0	0.990/0.990/0.990 ±0.003/±0.003/±0.003	0.990/0.990/0.990 ±0.003/±0.003/±0.003
AE	1	0.984/0.984/0.984 ±0.003/±0.003/±0.003	0.969/0.969/0.969 ±0.006/±0.006/±0.006	1/1/1 0/0/0	0.984/0.984/0.984 ±0.003/±0.003/±0.003	0.984/0.984/0.984 ±0.003/±0.003/±0.003
	2	0.987/0.987/0.987 ±0.003/±0.003/±0.003	0.975/0.975/0.975 ±0.006/±0.006/±0.006	1/1/1 0/0/0	0.987/0.987/0.987 ±0.003/±0.003/±0.003	0.987/0.987/0.987 ±0.003/±0.003/±0.003
	3	0.981/0.981/0.981 ±0.010/±0.010/±0.010	0.962/0.962/0.962 ±0.018/±0.018/±0.018	1/1/1 0/0/0	0.981/0.981/0.981 ±0.009/±0.009/±0.009	0.981/0.981/0.981 ±0.009/±0.009/±0.009

Table 10. OC-SVM and AE performance metrics in detecting ball defects.

Ball Faults (B007/B014/B021)						
Method	**Motor Load, hp**	**Performance Metrics**				
		Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
OC-SVM	1	0.856/0.981/0.993 ±0.024/±0.005/±0.004	0.986/0.986/0.986 ±0.007/±0.007/±0.007	0.783/0.975/1 ±0.029/±0.006/0	0.873/0.981/0.993 ±0.019/±0.005/±0.004	0.856/0.981/0.993 ±0.024/±0.005/±0.004
	2	0.665/0.993/0.979 ±0.056/±0.006/±0.004	0.985/0.985/0.985 ±0.012/±0.012/±0.012	0.603/1/0.974 ±0.046/0/±0.015	0.747/0.992/0.979 ±0.032/±0.006/±0.004	0.665/0.993/0.979 ±0.056/±0.006/±0.004
	3	0.855/0.993/0.993 ±0.013/±0.005/±0.005	0.986/0.986/0.986 ±0.010/±0.010/±0.010	0.781/1/1 ±0.016/0/0	0.872/0.993/0.993 ±0.011/±0.005/±0.005	0.855/0.993/0.993 ±0.013/±0.005/±0.005
AE	1	0.984/0.984/0.984 ±0.003/±0.003/±0.003	0.969/0.969/0.969 ±0.006/±0.006/±0.006	1/1/1 0/0/0	0.984/0.984/0.984 ±0.003/±0.003/±0.003	0.984/0.984/0.984 ±0.003/±0.003/±0.003
	2	0.987/0.987/0.987 ±0.003/±0.003/±0.003	0.975/0.975/0.975 ±0.006/±0.006/±0.006	1/1/1 0/0/0	0.987/0.987/0.987 ±0.003/±0.003/±0.003	0.987/0.987/0.987 ±0.003/±0.003/±0.003
	3	0.981/0.981/0.981 ±0.010/±0.010/±0.010	0.962/0.962/0.962 ±0.018/±0.018/±0.018	1/1/1 0/0/0	0.981/0.981/0.981 ±0.009/±0.009/±0.009	0.981/0.981/0.981 ±0.009/±0.009/±0.009

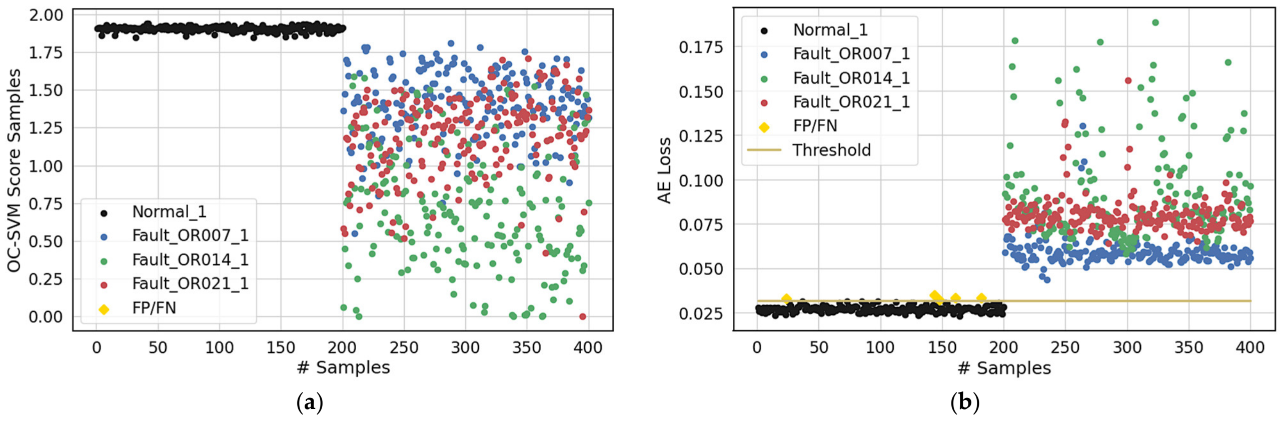


Figure 7. Anomaly scores of trained OC-SVM (**a**) and AE (**b**) models applied to test mixed data (normal + outer race fault modes) at 1-hp load.

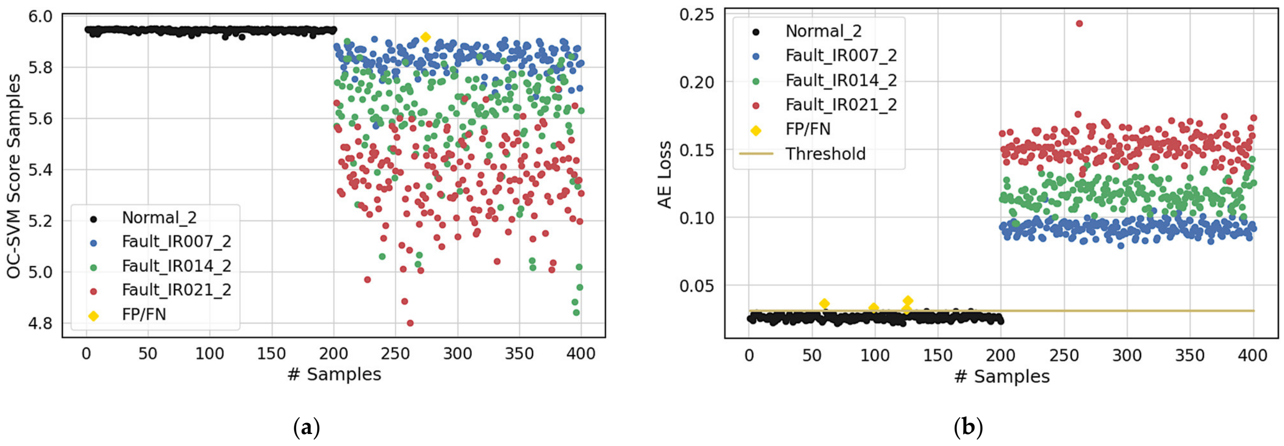


Figure 8. Anomaly scores of trained OC-SVM (**a**) and AE (**b**) models applied to test mixed data (normal + inner race fault modes) at 2-hp load.

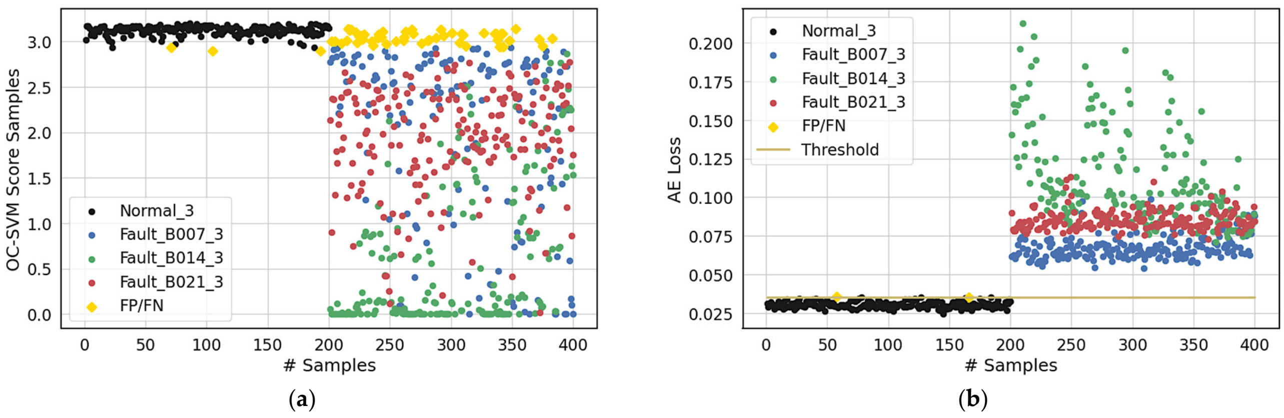


Figure 9. Anomaly scores of trained OC-SVM (**a**) and AE (**b**) models applied to test mixed data (normal + ball fault modes) at 3-hp load.

Confusion matrices in Figures 10–12 better detail the results in Tables 8–10. As expected, the worst case can be related to the OC-SVM method presenting many false negatives (FN = 51) when detecting the bearing ball fault of 0.007 inch -size. The resulting low recall value means the algorithm confuses the small-size ball defects with the normal mode.

From a detailed data analysis, both approaches are considered accurate in distinguishing high-severity faults from normal behavior. In Figure 8, the red dot corresponding to the highest AE loss (sample # 261—Fault_IR021_2) is the red dot with the most considerable distance to the OC-SVM score of the normal observations. The analysis in the time and frequency domain of sample # 261 in Figure 13 highlights high-amplitude vibration, mainly in the range of 3–4 kHz, for the 0.021-inch size fault class, much higher than vibration amplitudes shown by the same sample number corresponding to the anomaly classes with a lower fault size (Fault_IR014_2 and Fault_IR007_2). This evidence is also reflected by the values of the features given as input to the ML model and shown in Figure 14.

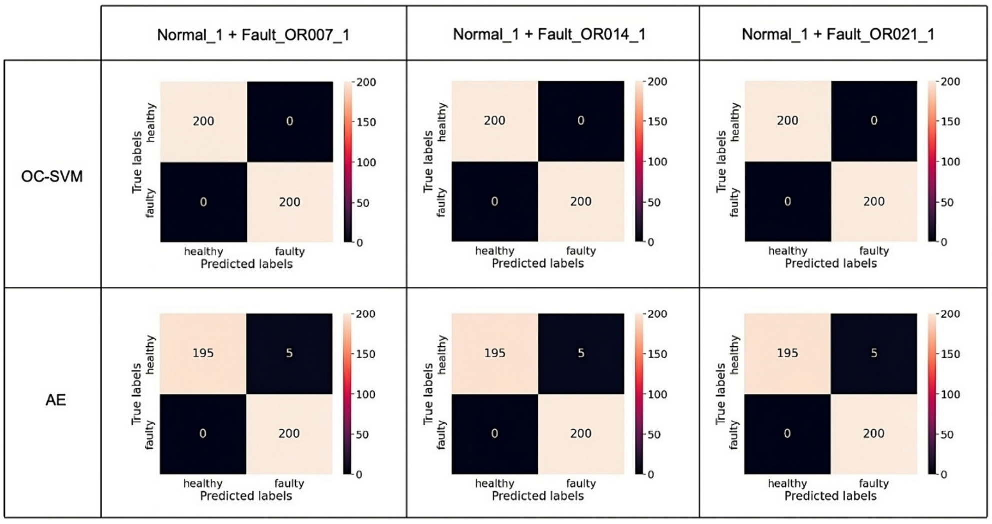


Figure 10. OC-SVM and AE confusion matrices on test dataset (normal + outer race fault modes) at 1-hp load.

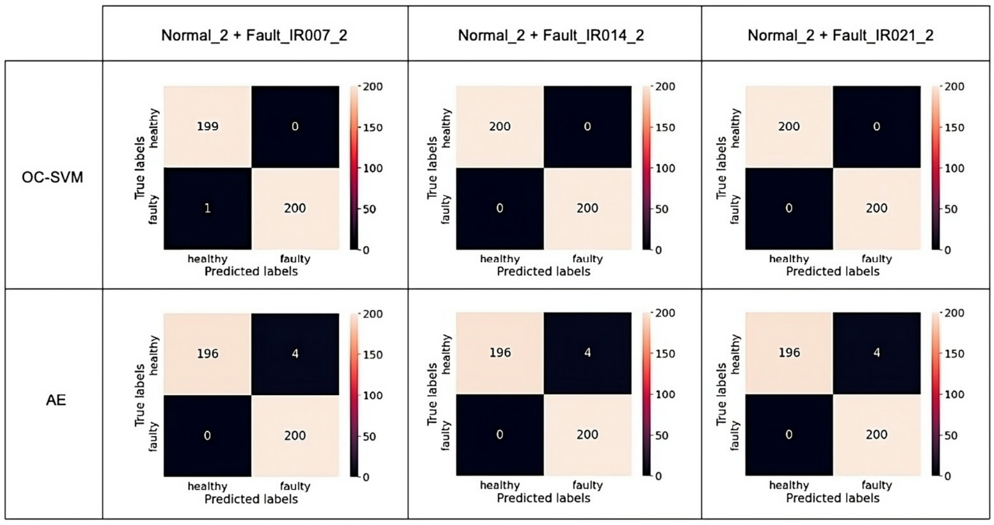


Figure 11. OC-SVM and AE confusion matrices on test dataset (normal + inner race fault modes) at 2-hp load.

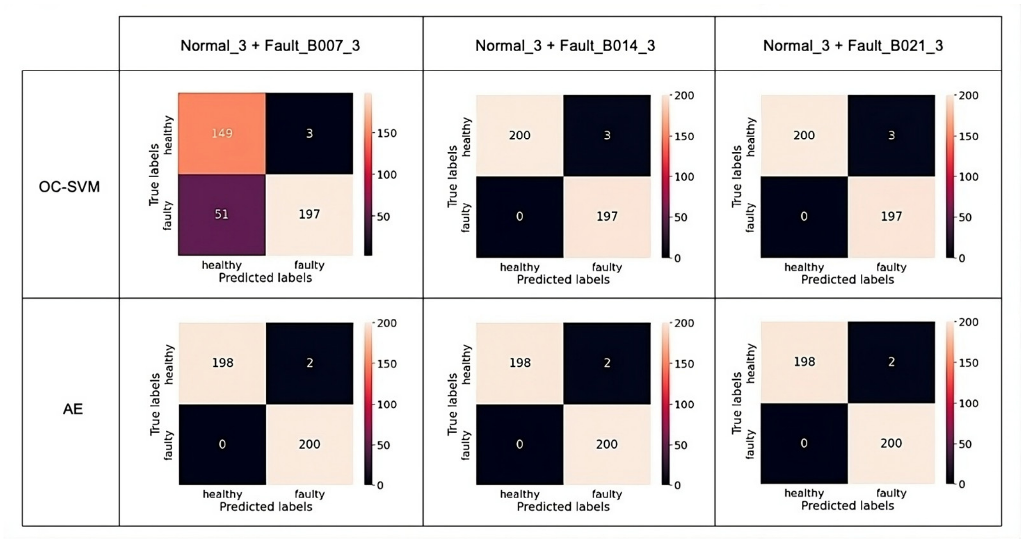
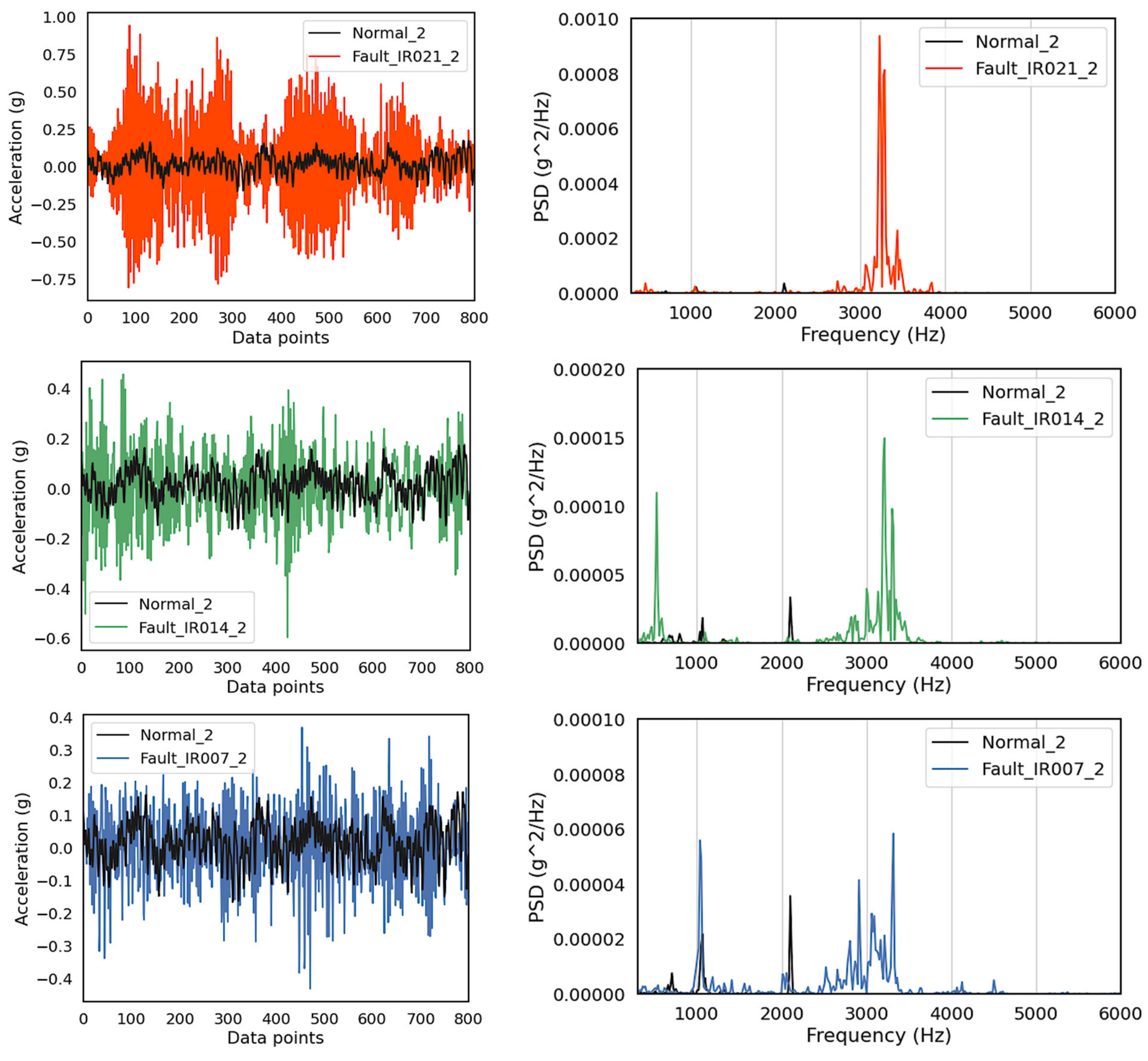


Figure 12. OC-SVM and AE confusion matrices on test dataset (normal + ball fault modes) at 3-hp load.



(**a**) (**b**)

Figure 13. Time (**a**) and frequency (**b**) analysis of sample # 261—normal and inner race fault signals at 2-hp load.

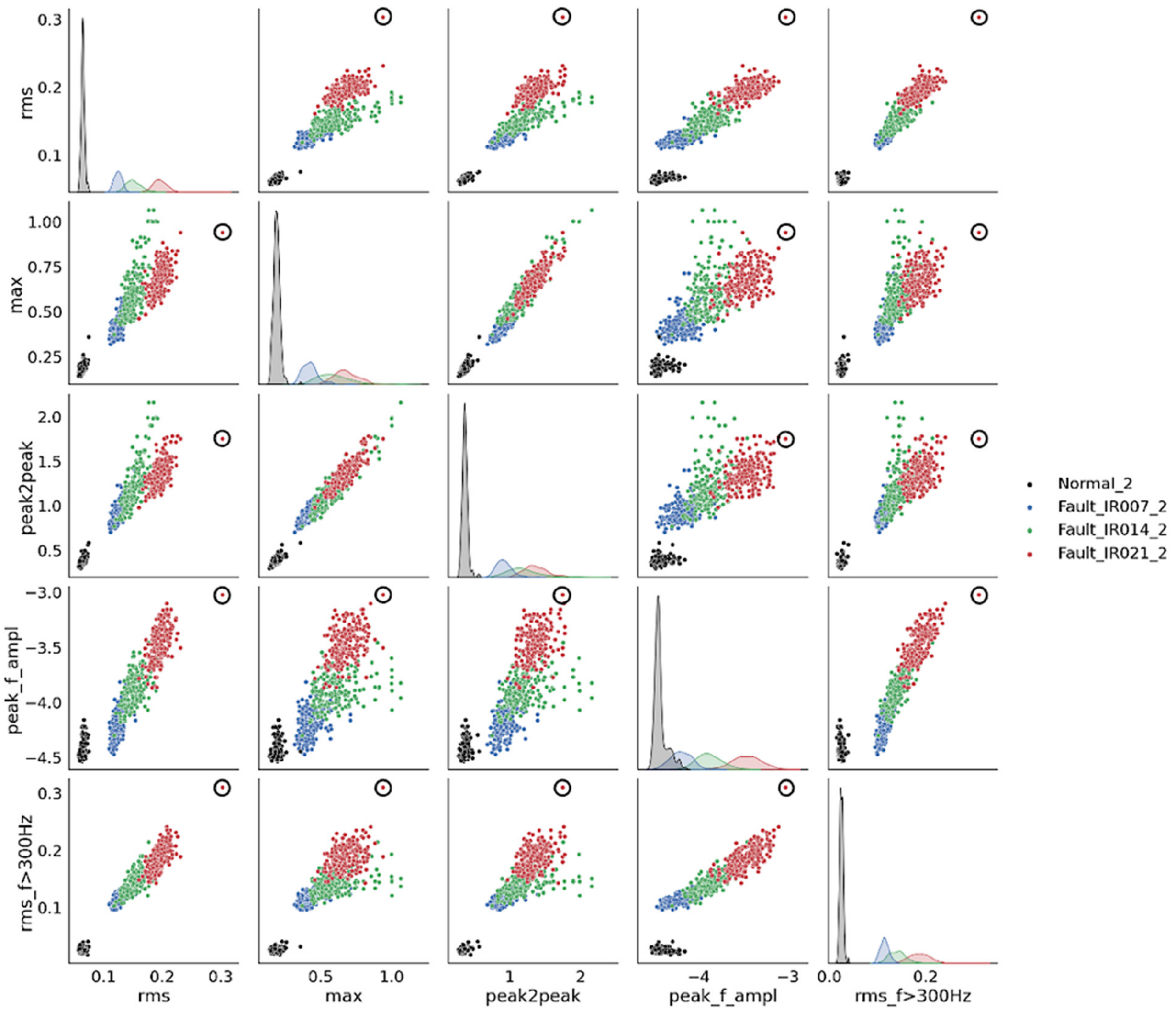


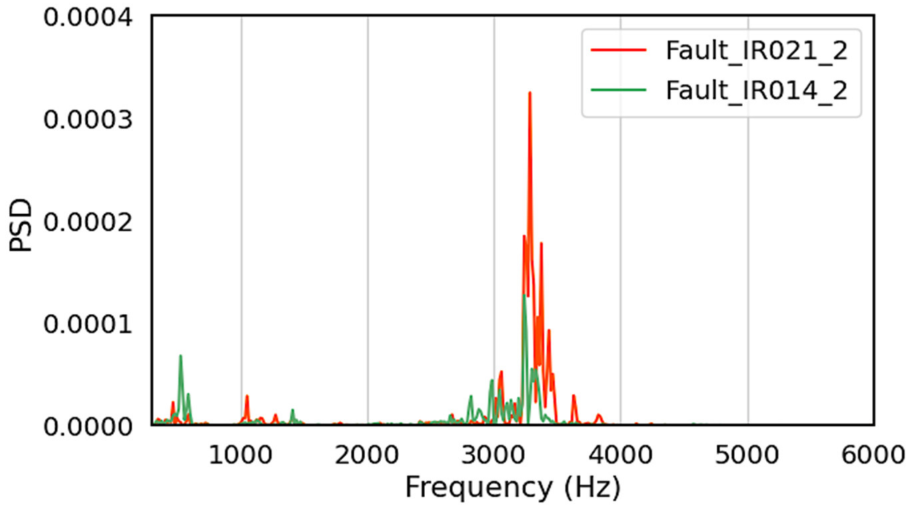
Figure 14. Scatter plots of test feature set for normal and inner race fault modes at 2-hp load.

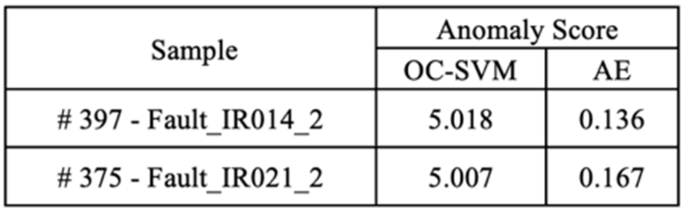
On the other hand, the ML algorithm's difficulty in correctly ranking the fault severity is highlighted. Figure 15 demonstrates differences in the 3–4 kHz amplitudes of the two compared signals, not detected by OC-SVM that assigns a similar AS to both samples. The AE reconstruction method, instead, is more effective in detecting even slight differences in the vibration signature. The scarce sensitivity of the OC-SVM method is probably due to the small set of features extracted, which can only capture some helpful information from the raw signals.

The unsupervised recognition of ball faults finally demonstrates the best overall performances of AEs-based methods. As reported in the literature [18,19], the ball faults are certainly the most difficult to diagnose, as in many cases, they do not give the classic spectrum symptoms when using established bearing diagnostic techniques. That explains the poor performance of OC-SVM in detecting small-size ball defects since it uses traditional time-domain and frequency-domain characteristic features.

So, the ML algorithm underperforms in comparison with the AE reconstruction method. Of course, the latter needs to be further improved. Varying model architecture regarding types of layers and hyperparameter settings during the training could reduce the false alarm rate of normal samples to zero.

Despite the lower performance, the strength of OC-SVM is the capability to provide additional information, giving insights into the possible root causes of the faults. As shown in Table 4, the SVM models are built on a reduced set of features, which differs for each analyzed anomaly class. That can help the anomaly diagnosis by identifying the features most sensitive to a specific type of fault. The real-time monitoring of their value ensures the detection and diagnosis of the fault, providing information about the defect's location. Evidence is given in Figure 16, where the inner race fault signature appears better described by peak amplitude. At the same time, kurtosis and RMS levels in the mid-frequency range characterize outer race and ball fault vibration, respectively.





Sample	Anomaly Score	
	OC-SVM	AE
# 397 - Fault_IR014_2	5.018	0.136
# 375 - Fault_IR021_2	5.007	0.167

Figure 15. Compared PSD spectra of normal and inner race fault signals at 2-hp load—Sample # 397 Fault_IR014_2, Sample # 375 Fault_IR021_2.

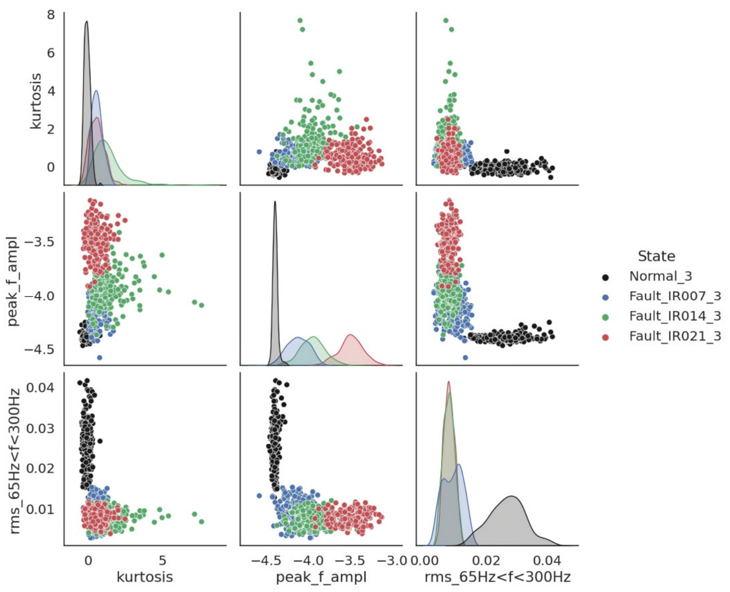


Figure 16. *Cont.*

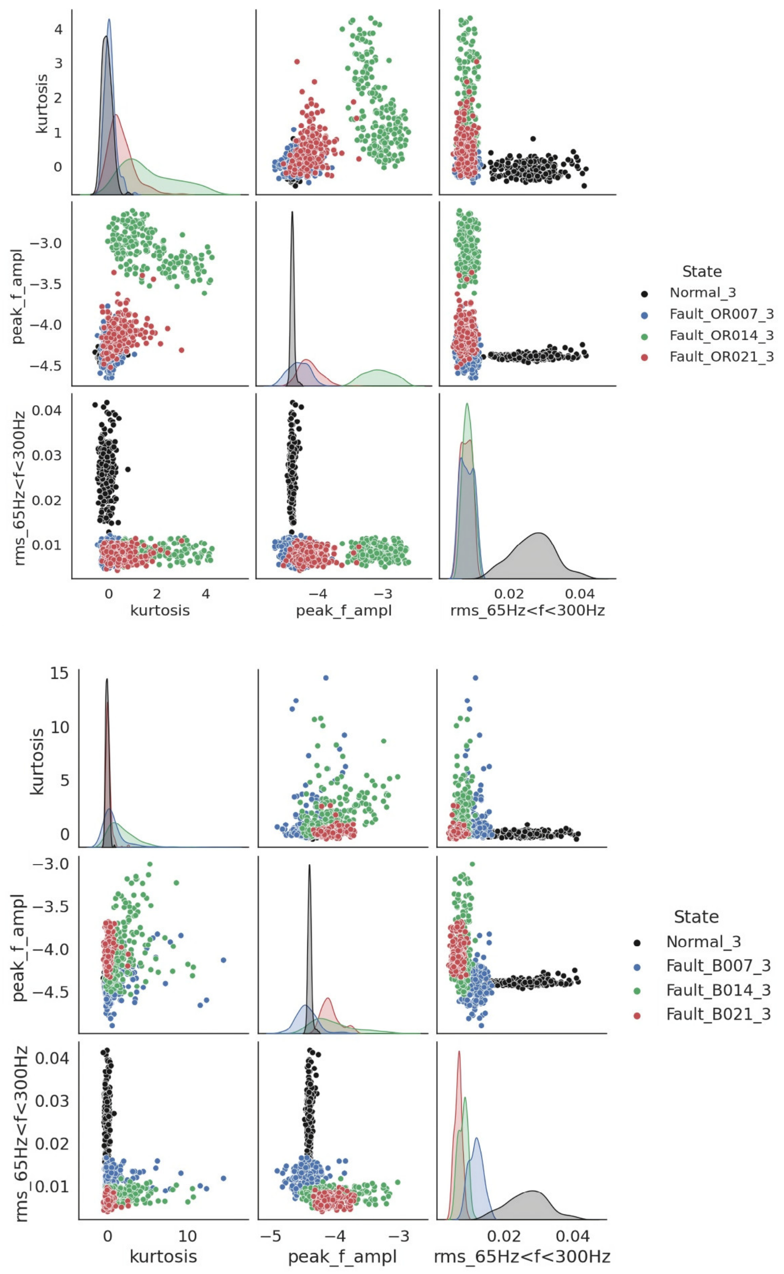


Figure 16. Scatter plots of test feature subset for normal and fault modes at 3-hp load.

5. Conclusions

The proposed research study aims to explore using two different unsupervised approaches for AD on a typical benchmark industrial dataset, i.e., the CWRU bearing fault dataset. The first approach is based on the manual extraction of typical vibration metrics to be provided as input to an ML model based on the OC-SVM algorithm. In contrast, the second DL approach automatically uses AEs to learn latent representation from raw data.

The evaluation metrics demonstrate that OC-SVM exhibits, in almost all cases, better performances than AEs in detecting bearing inner race and outer race defects, but it shows lower values of accuracy and recall for ball fault detection. Conversely, the approach based on AEs allows 100% recall for all experiments, presenting a 100% detection rate for all anomaly types, including the not easily diagnosable ball fault. Both approaches accurately distinguish high-severity faults from normal behavior, even though the ML algorithm presents a lower ability to rank fault severity properly.

So, the AE reconstruction method outperforms the ML algorithm. The main drawback of the DL method is the lack of interpretability of its internal operations, operating as a black box machine. On the other hand, depending on the anomaly class, selecting different feature sets allows identifying the most relevant features to detect a specific type of fault, improving the diagnosis results. That would make the process explainable to maintenance operators.

Future work may focus on improving the DL model to eliminate the false positive rate and studying methodologies to make the AE interpretable. The OC-SVM performances should also be improved to extend the number of extracted features, enabling the capture of all relevant and helpful information from the raw signals. Another development should make the feature selection process unrelated to the test set performance.

Author Contributions: Conceptualization, M.A.P., M.P. and M.E.; methodology, M.A.P., M.P. and M.E.; software, M.A.P.; validation, M.A.P.; formal analysis, M.A.P.; investigation, M.A.P.; data curation, M.A.P.; writing—original draft preparation, M.A.P.; writing—review and editing, M.A.P., M.P. and M.E.; visualization, M.A.P.; supervision, M.P. and M.E.; project administration, M.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: A publicly available dataset was analyzed in this study. The data can be found here: https://engineering.case.edu/bearingdatacenter/ (accessed on 17 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kamat, P.; Sugandhi, R. Anomaly Detection for Predictive Maintenance in Industry 4.0—A survey. *E3S Web Conf.* **2020**, *170*, 02007. [CrossRef]
2. Pimentel, M.A.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* **2014**, *99*, 215–249. [CrossRef]
3. Pota, M.; Russo, L.; Mancusi, E.; Crescitelli, S. Network of three catalytic reactors with periodical feed switching for methanol synthesis: Bifurcation analysis. *Comput. Aided Chem. Eng.* **2006**, *21*, 197–202. [CrossRef]
4. Carletti, M.; Masiero, C.; Beghi, A.; Susto, G.A. Explainable Machine Learning in Industry 4.0: Evaluating Feature Importance in Anomaly Detection to Enable Root Cause Analysis. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019. [CrossRef]
5. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv.* **2009**, *41*, 15. [CrossRef]
6. Bauw, M.; Velasco-Forero, S.; Angulo, J.; Adnet, C.; Airiau, O. From unsupervised to semi-supervised anomaly detection methods for HRRP targets. In Proceedings of the 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020; pp. 1–6. [CrossRef]
7. Riazi, M.; Zaiane, O.; Takeuchi, T.; Maltais, A.; Günther, J.; Lipsett, M. Detecting the Onset of Machine Failure Using Anomaly Detection Methods. In *Big Data Analytics and Knowledge Discovery*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 3–12. [CrossRef]
8. Widodo, A.; Yang, B.-S. Support vector machine in machine condition monitoring and fault diagnosis. *Mech. Syst. Signal Process.* **2007**, *21*, 2560–2574. [CrossRef]
9. Fernández-Francos, D.; Martínez-Rego, D.; Fontenla-Romero, O.; Alonso-Betanzos, A. Automatic bearing fault diagnosis based on one-class ν-SVM. *Comput. Ind. Eng.* **2013**, *64*, 357–365. [CrossRef]
10. Vos, K.; Peng, Z.; Jenkins, C.; Shahriar, R.; Borghesani, P.; Wang, W. Vibration-based anomaly detection using LSTM/SVM approaches. *Mech. Syst. Signal Process.* **2022**, *169*, 108752. [CrossRef]
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
12. Xia, M.; Li, T.; Liu, L.; Xu, L.; Silva, C.W.; de Silva, C.W. Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder. *IET Sci. Meas. Technol.* **2017**, *11*, 687–695. [CrossRef]
13. Principi, E.; Rossetti, D.; Squartini, S.; Piazza, F. Unsupervised electric motor fault detection by using deep autoencoders. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 441–451. [CrossRef]
14. Pota, M.; De Pietro, G.; Esposito, M. Real-time anomaly detection on time series of industrial furnaces: A comparison of autoencoder architectures. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106597. [CrossRef]
15. Huang, X.; Wen, G.; Dong, S.; Zhou, H.; Lei, Z.; Zhang, Z.; Chen, X. Memory Residual Regression Autoencoder for Bearing Fault Detection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3515512. [CrossRef]
16. Yoo, Y.; Jo, H.; Ban, S.-W. Lite and Efficient Deep Learning Model for Bearing Fault Diagnosis Using the CWRU Dataset. *Sensors* **2023**, *23*, 3157. [CrossRef] [PubMed]

17. Zhang, X.; Zhao, B.; Lin, Y. Machine Learning Based Bearing Fault Diagnosis Using the Case Western Reserve University Data: A Review. *IEEE Access* **2021**, *9*, 155598–155608. [CrossRef]
18. Smith, W.A.; Randall, R.B. Rolling element bearing diagnosis using the Case Western reserve university data: A benchmark study. *Mech. Syst. Signal Process.* **2015**, *64–65*, 100–131. [CrossRef]
19. Alonso-González, M.; Díaz, V.G.; Pérez, B.L.; G-Bustelo, B.C.P.; Anzola, J.P. Bearing Fault Diagnosis with Envelope Analysis and Machine Learning Approaches Using CWRU Dataset. *IEEE Access* **2023**, *11*, 57796–57805. [CrossRef]
20. Case Western Reserve University Bearing Data Center Website. Available online: https://engineering.case.edu/bearingdatacenter/ (accessed on 17 March 2023).
21. Zhang, W.; Peng, G.; Li, C.; Chen, Y.; Zhang, Z. A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals. *Sensors* **2017**, *17*, 425. [CrossRef]
22. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
23. Chollet, F. *Deep Learning with Python*; Simon and Schuster: New York, NY, USA, 2021.
24. Dong, G.; Huan, L. (Eds.) *Feature Engineering for Machine Learning and Data Analytics*; CRC Press: Boca Raton, FL, USA, 2018.
25. Christ, M.; Braun, N.; Neuffer, J.; Kempa-Liehr, A.W. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh—A Python package). *Neurocomputing* **2018**, *307*, 72–77. [CrossRef]
26. Jardine, A.K.; Lin, D.; Banjevic, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mech. Syst. Signal Process.* **2006**, *20*, 1483–1510. [CrossRef]
27. Gouriveau, R.; Medjaher, K.; Zerhouni, N. *From Prognostics and Health Systems Management to Predictive Maintenance 1: Monitoring and Prognostics*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
28. Correa, J.C.A.J.; Guzman, A.A.L. *Mechanical Vibrations and Condition Monitoring*; Academic Press: Cambridge, MA, USA, 2020. [CrossRef]
29. Brandt, A. *Noise and Vibration Analysis: Signal Analysis and Experimental Procedures*; John Wiley & Sons: Hoboken, NJ, USA, 2023.
30. Schölkopf, B.; Williamson, R.C.; Smola, A.; Shawe-Taylor, J.; Platt, J. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems 12.3*; MIT Press: Cambridge, MA, USA, 1999; pp. 582–588.
31. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [CrossRef] [PubMed]
32. Xiao, Y.; Wang, H.; Zhang, L.; Xu, W. Two methods of selecting Gaussian kernel parameters for one-class SVM and their application to fault detection. *Knowl.-Based Syst.* **2014**, *59*, 75–84. [CrossRef]
33. Schmidl, S.; Wenig, P.; Papenbrock, T. Anomaly detection in time series. *Proc. VLDB Endow.* **2022**, *15*, 1779–1797. [CrossRef]
34. Explorations in Artificial Intelligence and Machine Learning. Available online: https://www.routledge.com/rsc/downloads/AI_FreeBook.pdf (accessed on 16 May 2023).
35. Chollet, F. Keras. 2015. Available online: https://github.com/fchollet/keras (accessed on 24 March 2023).
36. Bakar, Z.; Mohemad, R.; Ahmad, A.; Deris, M.M. A Comparative Study for Outlier Detection Techniques in Data Mining. In Proceedings of the 2006 IEEE Conference on Cybernetics and Intelligent Systems, Bangkok, Thailand, 7–9 June 2006; pp. 1–6. [CrossRef]
37. Do, J.S.; Kareem, A.B.; Hur, J.-W. LSTM-Autoencoder for Vibration Anomaly Detection in Vertical Carousel Storage and Retrieval System (VCSRS). *Sensors* **2023**, *23*, 1009. [CrossRef] [PubMed]