POLITECNICO DI TORINO
Repository ISTITUZIONALE

Unsupervised Novelty Detection Methods Benchmarking with Wavelet Decomposition

Original
Unsupervised Novelty Detection Methods Benchmarking with Wavelet Decomposition / Albertin, U., Priarone, A., Cena, C., Martini, M., Chiaberge, M.. - ELETTRONICO. - (2024), pp. 198-207. (2024 8th International Conference on System Reliability and Safety (ICSRS) Sicily (Ita) 20-22 November 2024) [10.1109/icsrs63046.2024.10927428].

Availability:
This version is available at: 11583/2998715 since: 2025-04-24T08:03:29Z

Publisher:
IEEE

Published
DOI:10.1109/icsrs63046.2024.10927428

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

14 September 2026

Unsupervised Novelty Detection Methods Benchmarking with Wavelet Decomposition

Umberto Albertin[‡]
Department of Electronics and Telecommunications,
Politecnico di Torino
Torino, Italy
umberto.albertin@polito.it

Ariel Priarone[†]
Department of Electronics and Telecommunications,
Politecnico di Torino
Torino, Italy
ariel.priarone@polito.it

Carlo Cena[‡]
Department of Electronics and Telecommunications,
Politecnico di Torino
Torino, Italy
carlo.cena@polito.it

Mauro Martini
Department of Electronics and Telecommunications,
Politecnico di Torino
Torino, Italy
mauro.martini@polito.it

Marcello Chiaberge
Department of Electronics and Telecommunications,
Politecnico di Torino
Torino, Italy
marcello.chiaberge@polito.it

***Abstract*—Novelty detection is a critical task in various engineering fields. Numerous approaches to novelty detection rely on supervised or semi-supervised learning, which requires labelled datasets for training. However, acquiring labelled data, when feasible, can be expensive and time-consuming. For these reasons, unsupervised learning is a powerful alternative that allows performing novelty detection without needing labelled samples. In this study, numerous unsupervised machine learning algorithms for novelty detection are compared, highlighting their strengths and weaknesses in the context of vibration sensing. The proposed framework uses a continuous metric, unlike most traditional methods that merely flag anomalous samples without quantifying the degree of anomaly. Moreover, a new dataset is gathered from an actuator vibrating at specific frequencies to benchmark the algorithms and evaluate the framework. Novel conditions are introduced by altering the input wave signal. Our findings offer valuable insights into the adaptability and robustness of unsupervised learning techniques for real-world novelty detection applications.**

***Index Terms*—Unsupervised Machine Learning, Novelty Detection, Anomaly Detection, Predictive Maintenance, Artificial Intelligence, Neural Network, Prognostics, Autoencoder, K-Means, DBSCAN, Gaussian Mixture Model, One-Class Support Vector Machine, Isolation Forest, Local Outlier Factors**

I. Introduction

In a world driven by data, it is essential to identify unexpected events. Novelty detection, i.e. the process of identifying new or unknown data that deviate significantly from the expected or established patterns, is crucial in various fields, as it helps in identifying unusual occurrences that could indicate critical events such as system faults, fraud, or emerging trends.

[†] Corresponding author
[‡] This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 351/2022 and MUR – DM 117/2023.

For example, in manufacturing, novelty detection can be used to spot defects in the production plant [1], while in cybersecurity, it can help detect unusual patterns of behaviour that might indicate a security breach [2]. Lastly, in finance, anomaly detection can be vital for fraud detection, allowing institutions to prevent fraudulent transactions before they cause significant damage. Overall, the capability to recognize novelties enhances decision-making, improves safety, and reduces risks across various domains.

Traditional approaches to novelty detection often rely on statistical methods [3, 4]. These methods generally depend on predefined thresholds or models of normal behaviour, which can struggle in their ability to generalize to evolving patterns and unseen anomalies.

Artificial Intelligence (AI) has significantly advanced the field of novelty detection. Machine Learning (ML) models can learn from data and identify patterns that would be difficult to specify manually. Among AI approaches, supervised learning algorithms require labelled data to train models that can then predict anomalies. Notable examples are neural networks or support vector machines trained on historical data [5, 6]. Semi-supervised approaches, which use a combination of labelled and unlabelled data, offer a middle ground, as they leverage the available labelled data to improve the learning process while also incorporating unlabeled data to improve the model's robustness and accuracy [7, 8].

Unsupervised learning techniques, which do not require labelled data, are particularly valuable for novelty detection in situations where obtaining labelled data is impractical or impossible, due to a scarcity of data or to the costs associated with labelling it [7]. These methods include clustering algorithms, like K-means, autoencoders and density-based approaches [9, 10, 11, 12, 13, 14]. Unsupervised algorithms

can discover the underlying structure of the data and identify deviations. This capability makes unsupervised approaches versatile and powerful in changing environments.

To the best of our knowledge, the current literature lacks a comprehensive study which compares Unsupervised Machine Learning (UML) models able to produce a continuous degradation metric, with several preprocessing algorithms. We believe such a benchmark could provide valuable information to the novelty detection field, serving as a critical resource for researchers and practitioners.

In this study, we benchmarked various unsupervised AI algorithms for novelty detection on a dataset developed in a laboratory using a shaker to record vibrations at given frequencies, artificially generating novel conditions by changing the input wave signal. Our method aims at predicting a continuous metric rather than a binary label, as this approach allows us to evaluate the effectiveness and adaptability of these algorithms in identifying novel patterns in complex data structures. We extract a combination of statistical features and wavelet decomposition coefficients from the raw signal to use them as input for the UML models. Moreover, we defined a proper set of metrics to compare the performances of the framework configurations. Lastly, we analyze the performance of each configuration by recording its inference time to measure the computational effort required. The following sections detail our methodology, experiments, and results.

II. Related Works

Here we give an overview of previous works on novelty detection, focusing mainly on unsupervised approaches.

The K-means algorithm is used in [12, 13] to label the degradation states of bearings. In [12], the Intelligent Maintenance Systems (IMS) bearing dataset [15] is clustered using Traditional Statistical Features (TSF) and Mel-frequency Cepstral Coefficients, and the labelled time series are subsequently used to train a Convolutional Neural Network (CNN) recognition model using raw data, without the need to extract new features. Reference [13] computed TSF and Shannon's entropy to perform clustering on IMS and Case Western Reserve University (CWRU) [16] datasets. Pinedo et al. converted the time series into 2D representations, as images, using a CNN, i.e. Alexnet [17], to classify the degradation states.

Reference [10] proposed a method to quantify the health of bearings and validate it on the IMS dataset. They used time-domain features extracted from the vibration signal and applied a cross-correlation filter to remove the redundant features. Then, the K-means algorithm was used to select only the most relevant features through an optimization procedure aimed at obtaining the most dense and separated clusters. The Self-Organizing Map algorithm is then used to compute a health indicator for the bearing.

A method to efficiently initialize the cluster centres in the K-means algorithm is proposed in [18]. It merges the Ant Colony Optimization algorithm with K-means, and it has been validated by applying a three-layer Wavelet Packet Decomposition (WPD) on the CWRU dataset augmented using a sliding window method to prove the applicability of the method to large datasets.

In [11], the authors proposed a subset-based deep autoencoder model to learn discriminative features from datasets automatically. This approach has been validated on the CWRU, IMS, and Self-Priming Centrifugal Pump (SPCP) [19] bearing vibration datasets.

A case study on the IMS dataset is proposed in [14]. The authors leverage the knowledge of the fault frequencies of the bearings to attach labels to the data. The features considered initially are TSF and Redundant Second-Generation Wavelet Packaged Transform coefficients, though the dimensionality of the feature space is reduced by applying Principal Component Analysis (PCA). Lastly, K-means, Support Vector Machine (SVM) and agglomerative clustering are used to perform anomaly detection and to identify the failure modes.

Additionally, a powerful data stream classification approach is proposed in [9]. Notably, it is not a one-class approach, as it manages more than one "nominal" class, and deals with the emergence of novel classes during the classification task by adding, to the training set, clusters of new data points with a given level of cohesion.

Reference [1] presents a machine learning framework for real-time anomaly detection in sensor data, aiding companies in creating datasets for predictive maintenance. It details an optimized software architecture for efficient novelty detection, validated through a digital model and real-world case studies.

Finally, a computer vision method to detect anomalies in mechanical systems is proposed in [20]. With respect to the previous methods, it has the advantage of evaluating vibrations in multiple points of interest without physical contact with the observed components.

Another model used in the context of novelty detection is the Local Outlier Factor (LOF) [21], whose continuous metric depends on the position of the point and the density of known points around it. This concept has been extended to consider also the clustered structure of the data in [22], which defines the metric Cluster-Based LOF (CBLOF).

Moreover, a simple way of quantifying the normality of data using a distance metric is proposed in [23]. The authors propose to compute a novelty score, that is the distance of the record from the nearest cluster, normalized by the standard deviation of the distance of the known points in the cluster. This method has been tested on vibration data collected on a jet engine.

Another distance-based method is proposed in [24], as it provides a hard classification of the data as normal, extension, or unknown based on the radius of the closest cluster and the position of the point to be evaluated. If the point is outside the decision boundary but within a tolerance, then it is classified as an extension, and the learned model is updated. If the point is outside the tolerance, then it is classified as unknown. Unknown points are kept in a buffer and used to update the model when new classes emerge within the data.

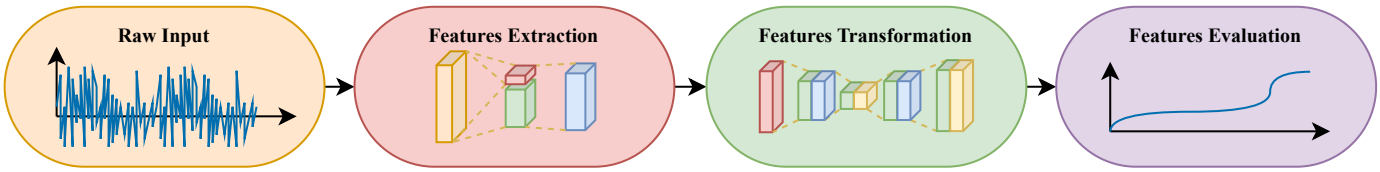

Fig. 1. The proposed framework architecture consists of three main blocks. The feature extraction block uses WPD to extract wavelet coefficients. These coefficients, along with statistical measures, form the output of this block. The feature transformation block employs either Autoencoder or PCA to transform the extracted features, enabling the detection of different behaviours in the novelty metric computation. Finally, the feature evaluation block uses six unsupervised machine learning models to compute the novelty metric of the transformed features.

III. Methodology

A. Novelty detection framework architecture

The proposed novelty detection framework is unsupervised. The training data used reflects the system's normal behaviour without any labels. Unlike classification algorithms, our framework does not have information about the system's modes. The trained model is then used to detect anomalies in the incoming data, assessing the deviations from the nominal behaviour. The proposed architecture is shown in Fig. 1.

The raw input are the time series collected by an accelerometer sensor. These data are then extracted into a set of key features. These features are then transformed into a latent space that can have lower or higher dimensions. Finally, the features are evaluated with several UML models producing a novelty metric (NM) (i.e. the framework's output). The advantage of this framework is related to provide deviation's severity instead of setting a binary flag (i.e. 0,1) for normal and anomalous behaviours, which is the common approach in anomaly detection tasks. Each block is better explained in the following section.

B. Features Extraction

The features extracted from the time series are a combination of statistical and WPD coefficients. The statistical features extracted for each signal are: mean, root-mean-square (RMS), peak-to-peak (P2P), standard deviation (STD), skewness and kurtosis.

The WPD is a tree-based decomposition method able to extract the frequency content of a signal. In this work, a Daubechies wavelet is used. The decomposition method divides the input data $x \in \mathbb{R}^N$ into two subsets: a set with high-frequency and the other with low-frequency content, each containing $N/2$ elements. This process can be applied an arbitrary number of times (up to $\log_2(N)$) for each subset created. The process is repeated recursively until the desired number of levels L is reached. Hence, the final output is a signal decomposed into 2^L vectors in which each vector has $N/2^L$ elements. Finally, each vector generated by the decomposition is transformed into a single value using the l_2 norm. This value assumes the role of a feature. The 2^L computed norms form the WPD feature array. The structure of the decomposition is shown in Fig.2.

In the end, the WPD feature vector is concatenated with the six statistical features to form the complete features array

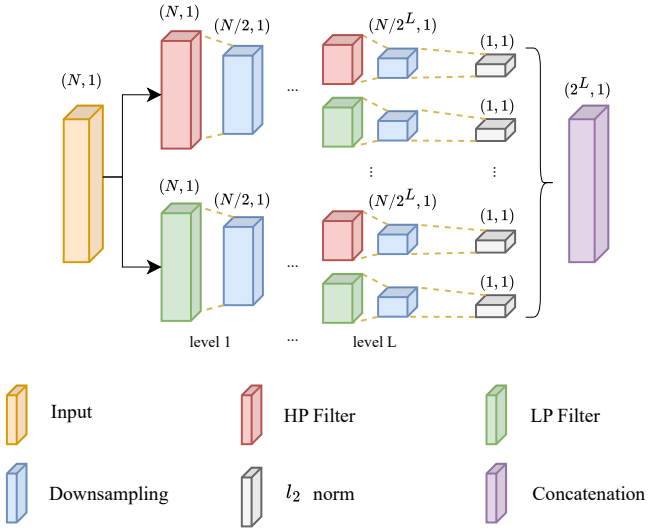

Fig. 2. Wavelet Packet Decomposition features extraction architecture. At each level of decomposition, the input signal is split into two sub-band signals, each with a dimension of $N/2$. This process is repeated through L levels, resulting in 2^L sub-band signals. For each sub-band signal, the l_2 norm is computed, condensing each array into a single value. These values are then aggregated into a final array of dimension 2^L, forming the complete WPD feature array that encapsulates the characteristics of the original time series data.

with dimension $(2^L + 6, 1)$ representing the final output of the features extraction block.

After the feature extraction process, the data are normalized, along the whole training dataset, removing the mean and scaling to unit variance in order to use them for the following steps.

At this stage, the extracted features can be directly used to train the models. However, additional processing can be implemented to analyze the different behaviours of the unsupervised model. Specifically, two different Autoencoders and PCA are tested to observe the change in the novelty metric, as explained in the next section.

C. Features Transformation

In this section, the feature transformation block is explained. Three different algorithms, an undercomplete and an overcomplete autoencoder, and PCA are considered.

1) Autoencoder: The first proposed architecture contains an autoencoder (AE), a generative neural network (NN) composed of an encoder that increases or reduces the input

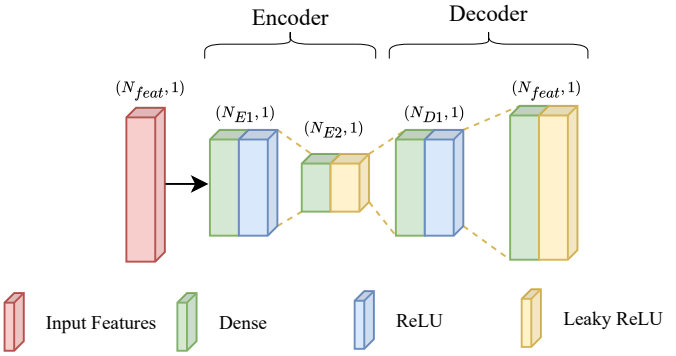

Fig. 3. Architecture of the proposed undercomplete autoencoder feature transformation algorithm. The NN model is composed of fully connected layers with the following shape: $N_{E2} < N_{E1} < N_{feat}$ and $N_{E2} < N_{D1} < N_{feat}$. The input is the preprocessed data obtained after the statistical and wavelet transformation.

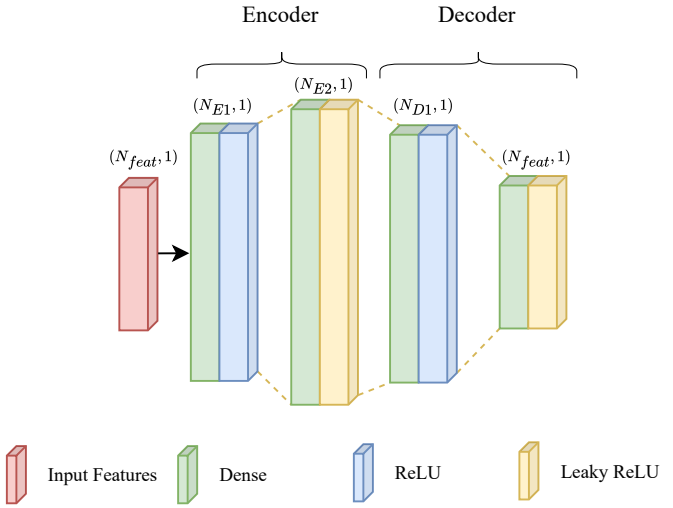

Fig. 4. Architecture of the proposed overcomplete autoencoder feature transformation algorithm. The NN model is composed of fully connected layers with the following shape: $N_{feat} < N_{E1} < N_{E2}$ and $N_{feat} < N_{D1} < N_{E2}$.

dimensionality and a decoder that tries to reconstruct the original input. In this kind of model, the number of neurons in the input and in the output layers must be the same, while the hidden layers can vary in shape and size. The autoencoder is considered undercomplete (AER) if the latent space has a lower dimension than the input and overcomplete (AEA) if the latent space has a higher dimension than the input. Formally, let $X \in \mathbb{R}^n$ represents the input data, n represents the input dimension, the encoder $f_{enc} : \mathbb{R}^n \rightarrow \mathbb{R}^p$, and the decoder $f_{dec} : \mathbb{R}^p \rightarrow \mathbb{R}^n$. The autoencoder can be represented as $X_{recon} = f_{dec}(h)$ where h is the latent space $h = f_{enc}(X)$. Hence, the complete form of the eutoencoder can be represented as $X_{recon} = f_{dec}(f_{enc}(X))$. The undercomplete autoencoder has the latent space dimension p lower than the input ($p < n$), while the overcomplete autoencoder has a higher dimension ($p > n$). Both NN models have two layers for the encoder and two layers for the decoder.

a) Undercomplete Autoencoder: The shape of the undercomplete autoencoder is shown in Figure 3. The input data are the features $n = N_{feat} = 2^L + 6$ explained and extracted in Section III-B. The layers have the following dimensions: N_{E1}, N_{E2}, N_{D1} where $N_{E2} = p$ and $N_{E2} < N_{E1} < N_{feat}$. Then, the decoder transforms the latent space into the original dimension N_{feat}, where $N_{E2} < N_{D1} < N_{feat}$. The first layer for both the encoder and decoder uses a ReLU activation function, while the second layer uses a LeakyReLU activation function to enhance the model's robustness and prevent data saturation to zero. The model employs Mean Squared Error (MSE) as the loss function.

b) Overcomplete Autoencoder: The same considerations are performed also for the overcomplete autoencoder shown in Figure 4. In this configuration, the input size is increased in the hidden layers of the autoencoder. Formally, the encoder is structured such that $N_{feat} < N_{E1} < N_{E2}$, while the decoder follows $N_{feat} < N_{D1} < N_{E2}$. Both the encoder and decoder employ ReLU activation in the first layer and LeakyReLU activation in the second layer, with MSE as the loss function, similar to the undercomplete autoencoder.

2) Principal Component Analysis: Principal Component Analysis is employed to transform the features array into a lower dimensional feature space preserving the input variance. The number of features is reduced as a function of the ratio of variance to be preserved.

Formally, let $X \in \mathbb{R}^n$ represents the input data, where $n = N_{feat}$. The PCA function is $h = PCA(X)$, where h is the latent space with reduced features $PCA : \mathbb{R}^{N_{feat}} \rightarrow \mathbb{R}^{N_{PCA}}$ with $N_{PCA} < N_{feat}$.

The three architectures (i.e. AEA, AER, PCA) provide different numbers of features as input to the unsupervised models which produce as output the novelty metric value better explained in the next section.

D. Features Evaluation

Here, the six unsupervised models that can be implemented in combination with all the previous feature transformation methods (sec. III-C) are explained in detail. The input of the unsupervised models are the features transformed by the autoencoders or by the PCA. The output of this block is the novelty metric. When the autoencoders are used, the features are the latent output of the last layer of the encoder, in our case E_2.

1) K-Means: The K-Means algorithm is a simple and widely used clustering algorithm. The algorithm requires as input the number of clusters to be created, and the data to be clustered. The algorithm first initializes k centroids and then updates them iteratively, to better fit the data. In this paper, the variant "K-means++" [25] is used.

The algorithm is trained multiple times with different numbers of clusters to select the best one. The silhouette score is the metric used to choose the number of cluster. Formally, the silhouette score [26] $s(I)$ of a sample I is a measure of how

well the sample fits in its cluster, compared to other clusters, and is defined as:

$$s(I) = \frac{b(I) - a(I)}{\max(a(I), b(I))} \tag{1}$$

$$a(I) = \frac{1}{|C_i| - 1} \sum_{J \in C_i, I \neq J} d(I, J) \tag{2}$$

$$b(I) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{J \in C_k} d(I, J) \tag{3}$$

where C_i is the cluster assigned to sample I, $d(I, J)$ is the distance between samples I and J, C_k is another cluster, $|C_i|$ is the number of samples in cluster C_i, and $|C_k|$ is the number of samples in cluster C_k. Hence, $a(I)$ is the average distance between sample I and all other samples in the same cluster, and $b(I)$ is the average distance between sample I and all samples in the nearest cluster. The best number of clusters is selected on the training that achieves the highest average silhouette score across all samples.

Once the model is trained, the centroids are known and a way of measuring how novel the new data are is needed. In this paper, for the k-means models, we propose a Novelty Metric similar to the one in [23]. The approach involves computing the distance from a new sample to the closest cluster's centroid, comparing this distance with the cluster's radius, and normalizing by this radius (instead of by the standard deviation of the samples in the cluster, as done in [23]). The NM is defined as:

$$\text{NM}_{\text{KMeans}}(I) = \frac{d(I, c_i) - r_i}{r_i}, \quad r_i = \max_{J \in C_i} d(J, c_i) \tag{4}$$

where I is the sample to be evaluated, c_i is the closest centroid to I and r_i is the radius of the closest cluster. This normalization accounts for the cluster size while preserving the property of the NM being negative for samples strictly inside the clusters, zero for samples on the boundary of a cluster, and positive for novel samples.

2) DBSCAN: Another clustering algorithm tested is the DBSCAN [27]. It works by dividing the data into "core" and "reachable" points, based on the minimum cluster size MinPts and the search radius ε. It has the ability to discard some samples in the training data as noise. In this case, the novelty metric for a new sample I is defined as follows:

$$\text{NM}_{\text{DBSCAN}}(I) = \frac{1}{\varepsilon} \min_{J \in D} d(I, J) \tag{5}$$

Where D is the set of samples in the train data that are not marked as noise.

3) GMM: The Gaussian Mixture Model (GMM) is a probabilistic model that assumes that the data is generated from a mixture of several Gaussian distributions. It estimates the parameters of each distribution. In this case, the novelty metric is the likelihood that a given sample belongs to the distribution mixture.

4) nuSVM: The one-class Support Vector Machine (nuSVM) is a variant of the standard SVM that applies a transformation to the input space in order to separate the data from the origin. New data samples that appear closer to the origin than the training samples are considered outliers.

In this case, we consider the distance of the sample from the separating hyperplane, as proposed by [28], as novelty metric $\text{NM}_{\text{nuSVM}}(i)$.

5) IF: The Isolation Forest (IF) is an ensemble algorithm that trains a set of decision trees to isolate the outliers. The sample is compared with a threshold at each node of the tree, until a leaf is reached. An outlier is expected to reach a leaf in a few steps because it is dissimilar to known samples, while an inlier is expected to reach a leaf after many steps, because it is similar to known samples. We use as novelty metric $\text{NM}_{\text{IF}}(i)$ the "anomaly score" defined by the authors [29].

6) LOF: The Local Outlier Factor (LOF) is a density-based algorithm that computes the local density of a sample compared to the local density of its neighbours. The algorithm then returns a metric that indicates "the degree of being an outlier" [30]. We consider the result of the algorithm as the $\text{NM}_{\text{LOF}}(i)$.

IV. Experiments

A. Dataset acquisition

A pivotal point of our research is the realization of a dataset to test the Novelty Detection algorithms in a real environment. For this purpose, a shaker (mod. n° K2007E01) and an accelerometer (mod. n° STM32L4R9, an evaluation board of ST microelectronics equipped with an accelerometer sensor) were used to collect a vibration dataset. The evaluation board is mounted on the shaker to gather the dataset with reduced noise and uncertainties.

The shaker is controlled by a PC, reproducing an audio file. Two audio files have been generated, the first reproduces a signal $v_1(t)$ and the second one the signal $v_2(t)$ that we define as:

$$v_1(t) = \sum_{i=1}^{9} A_i \cdot \sin(2\pi \cdot \alpha_i \cdot t)) \tag{6}$$

$$A = \{0.5, 0.5, 0.5, 1, 1, 1, 0.5, 0.5, 0.5\}$$
$$\alpha = \{50, 100, 150, 230, 300, 440, 460, 530, 600\}$$

$$v_2(t) = \sum_{i=1}^{9} B_i \cdot \sin(2\pi \cdot \beta_i \cdot t)) \tag{7}$$

$$B = [0.5, 0.5, 0.5, 1, 0.2, 1, 0.5, 0.5, 2]$$
$$\beta = [50, 100, 150, 230, 300, 440, 460, 530, 600]$$

where A, B are the weights and α, β are the frequencies of the harmonics. The volume setting is used to change the $P2P$ amplitude of the physical signal generated by the PC.

Firstly, the signal v_1 was fed to the shaker, with five different volume settings, acquiring five sets of data (set 1 - set 5). Every time the volume was changed, the new $P2P$ value was

TABLE I
EXPERIMENTAL SETUP FOR DATA COLLECTION.

Set name	Signal Type	P2P [V]
Set 1	$v_1(t)$	0.25
Set 2	$v_1(t)$	0.50
Set 3	$v_1(t)$	0.75
Set 4	$v_1(t)$	1.00
Set 5	$v_1(t)$	1.25
Set 6	$v_2(t)$	0.50
Set 7	$v_2(t)$	0.75
Set 8	$v_2(t)$	1.00

measured with the oscilloscope. Lastly, this procedure was repeated using the signal v_2, to record three more sets (set 6 - set 8). Each of the eight sets of vibration data acquired consists of a record that lasts 206 seconds and has been split into 1-second chunks to generate 206 records. The setup used to collect the eight sets of vibration data is resumed in Table I.

The highest frequency chosen is 600Hz, hence the sampling frequency of the evaluation board is set to $f_s = 1666$Hz respecting the Nyquist–Shannon sampling theorem.

B. Features extraction and normalization

The procedure described in Section III-B is applied to the experimental data collected. The depth level L of the WPD decomposition tree is chosen to be 6, obtaining a WPD feature array with 64 elements. The 6 statistical features are also computed, yielding a final feature array with 70 elements.

C. Models comparison

The performances of each architecture need to be evaluated in order to perform the benchmark analysis for each architecture. Four different performance metrics are computed for each framework.

1) Variance: The variance of the novelty metric is evaluated using the remaining part of the nominal data not used for the training. A good model should produce a stable novelty metric when the data represent the same normal conditions. High variance means that the model is either too sensitive to noise or cannot generalize normal behaviour well. Formally, this metric is defined as:

$$Variance = \frac{1}{n}\sum_{i=1}^{n}\left(\text{NM}(i) - \overline{\text{NM}}\right)^2 \tag{8}$$

where $\text{NM}(i)$ is the novelty metric of the i-th normal sample, $\overline{\text{NM}}$ is the mean of the novelty metric along all the normal samples, and n is the number of samples.

2) Reactivity: This metric is used to compare different models under the same conditions. The reactivity of the model is defined as the difference between the mean nominal and worst novelty metric. High reactivity indicates that the model can better distinguish between nominal and novelty samples. Formally, the reactivity is defined as:

$$\text{Reactivity} = \frac{1}{n}\sum_{i=1}^{n}\text{NM}(i) - \frac{1}{m}\sum_{j=1}^{m}\text{NM}(j) \tag{9}$$

where $\text{NM}(i)$ is the novelty metric of the i-th nominal sample and $\text{NM}(j)$ is the novelty metric of the j-th novel sample.

3) Inference Time: The inference time to evaluate a single sample (i.e. the time interval used by the UML model to compute the NM). This metric is averaged by evaluating several samples. The inference time is measured using an Intel CPU i9-12900K.

4) Feature Percentage: The Feature Percentage (FP) indicates the percentage of dimensionality preserved after the feature transformation with respect to the original ones. An FP of 100% means that all original features are retained. A lower FP is preferable for faster computations and to avoid the challenges posed by high-dimensional data (the curse of dimensionality).

D. Models Optimization

The algorithms, described in section III-C, are also optimized to select the best-performing one for each UML model. Concerning the autoencoder, the optimization process is performed by changing the number of neurons of the encoder N_{E1} which is the same for the decoder $N_{E1} = N_{D1}$, the number of neurons of the latent space of the encoder N_{E2}, the learning rate, and the batch size. For the PCA, the only parameter changed during the optimization is the number of latent features n_f. The variance is chosen as objective function because, at the time of optimization, it is the only variable available without knowledge of the future of the system (e.g. reactivity). Minimizing this variance is crucial because it reflects the algorithm's ability to model a nominal condition of the system. A lower variance indicates better performance because it suggests that the model is less susceptible to noise or environmental changes during the normal operation of the system.

The optimization used is a Gaussian process-based Bayesian algorithm. It iteratively suggests values for the hyperparameters that optimize the objective function, aiming to improve the model's robustness and detection capabilities.

The first 100 samples of set 1 of Table I are used for training, while the remaining 106 samples of the same set are used to compute the variance metric. Formally:

$$J = min(\sigma_{100}NM)$$
$$\sigma_{100}(NM) \rightarrow N_{E1,D1}, N_{E2}, lr, bs, n_f$$

where J is the objective function to minimize, lr is the learning rate, bs is the batch size, and n_f is the number of PCA's latent features. The sampler needs a range of values to optimize in order to reach the minimum variance. The ranges chosen for each variable are reported in Table II.

The parameters obtained by applying the optimization process are reported in Table III.

E. Model evaluation

Performance metrics are computed across all combinations of UML models and feature transformation methods.

The first 100 samples of set 1 are used to train the UML models. Four combinations are benchmarked for each UML

TABLE II
HYPERPARAMETERS RANGES CHOSEN FOR THE MODELS' OPTIMIZATION ALGORITHM

Transf. Method	$\mathbf{N_{E1}}, \mathbf{N_{D1}}$	$\mathbf{N_{E2}}$	**lr**	**bs**	$\mathbf{n_f}$
AEA	$50-65$	$10-45$	$0.01-0.1$	$32, 64$	-
AER	$75-80$	$85-100$	$0.01-0.1$	$32, 64$	-
PCA	-	-	-	-	$2-70$

TABLE III
TUNED HYPERPARAMETERS. AEA IS THE OVERCOMPLETE AUTOENCODER, AER IS THE UNDERCOMPLETE AUTOENCODER

Model	$\mathbf{N_{E1}}, \mathbf{N_{D1}}$		$\mathbf{N_{E2}}$		**lr**		**bs**		$\mathbf{n_f}$
	AEA	AER	AEA	AER	AEA	AER	AEA	AER	PCA
KMeans	80	61	85	32	0.08	0.03	64	64	3
DBSCAN	75	56	85	10	0.20	0.08	32	64	2
GMM	79	61	85	10	0.08	0.01	32	64	2
nuSVM	80	65	93	10	0.09	0.10	32	64	2
IForest	79	50	85	45	0.02	0.03	64	64	70
LOF	79	50	88	10	0.03	0.01	32	32	3

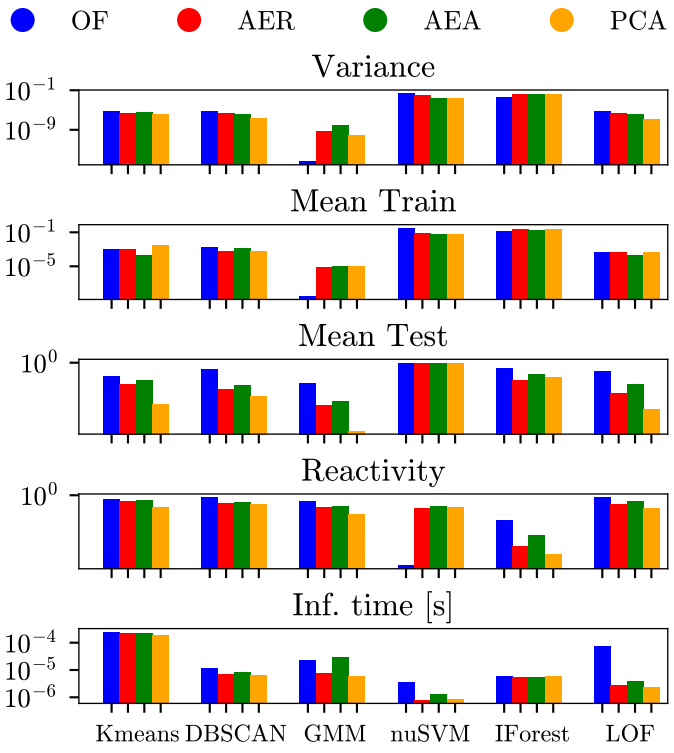

Fig. 5. Graphical representation of performance metrics reported on Table IV. All the y axes are displayed in a logarithmic scale. AEA is the overcomplete autoencoder, AER is the undercomplete autoencoder and OF is the original features.

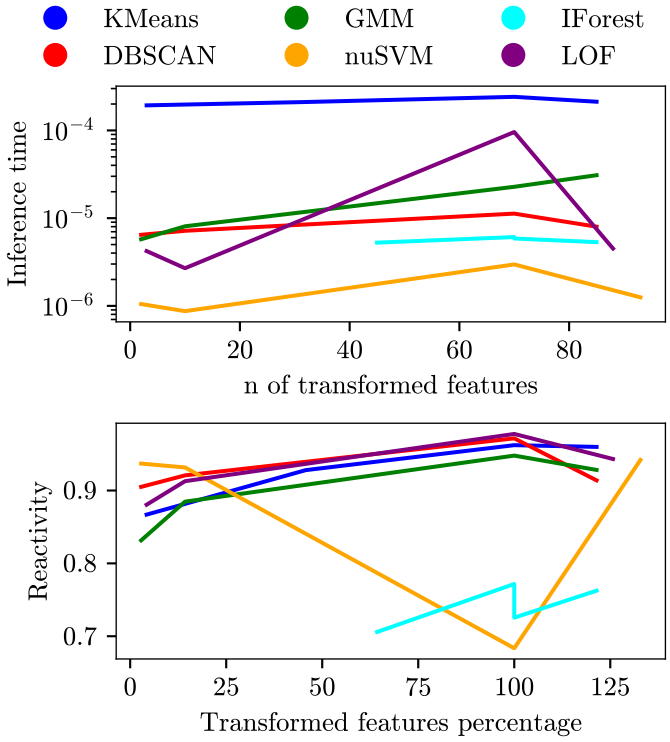

Fig. 6. In the upper plot, the correlation between the inference time and the number of features. In the bottom plot, the correlation between the reduced feature dimensionality ratio FP and the reactivity.

model: a test is performed without any features transformation using only the features extracted by the features extraction block (we refer to this scenario with OF, which stands for "Original Features"). The other three use the undercomplete and overcomplete AE, and the PCA as transformation algorithms.

The remaining 106 samples of set 1 are used to compute the variance and the mean of the nominal signal (i.e. "mean - nominal signal".

The 206 samples of Set 5, are used as a reference test to compute the "mean - novelty signal" and consequently the reactivity. Set 5 is selected as the reference novelty test because it represents the most dissimilar signal among the sets.

The novelty metric produced by all models is computed and normalized to a common scale where $\min(\text{NM}) = 0$ and $\max(\text{NM}) = 1$ for a fair comparison.

The resulting performance metrics are reported in Table IV, where the best transformation method for each UML model, for each metric, is highlighted in bold. Additionally, the best results among all the UML models and transformation combinations are highlighted with an asterisk, for each metric. The same results are also visually depicted in Figure 5 for clearness.

Looking at Table IV, we make some considerations about the best UML models and the best transformation methods.

Regarding the dimensionality reduction of the feature space, the PCA is almost always the transformation that selects the

TABLE IV

PERFORMANCE METRICS FOR ALL THE PREPROCESSING AND MODEL COMBINATIONS. OF MEANS ORIGINAL FEATURES, AER MEANS AUTOENCODER REDUCED, AEA MEANS AUTOENCODER AUGMENTED. n_f IS THE NUMBER OF FEATURES TRANSFORMED BY THE AUTOENCODERS OR PCA. FP IS THE FEATURES PERCENTAGE WITH RESPECT TO THE ORIGINAL DIMENSION. VARIANCE DEFINED IN THE EQUATION 8, AND THE REACTIVITY DEFINED IN EQUATION 9. TRAIN SIGNAL IS SET 1 OF TABLE I AND TEST SIGNAL IS SET 5. THE SYMBOLS ⇓ AND ⇑ INDICATE IF THE VALUE SHOULD IDEALLY BE MINIMIZED OR MAXIMIZED, RESPECTIVELY. THE BEST RESULTS AMONG ALL THE UML MODELS AND FEATURES TRANSFORMATION COMBINATIONS ARE HIGHLIGHTED WITH *, FOR EACH METRIC.

UML model	Transformation	n_f	FP [%] ⇓	Variance [10^{-3}] ⇓	Mean		Reactivity ⇑	Inference Time [μs] ⇓
					Nominal signal ⇓	*Novelty Signal* ⇑		
KMeans	OF	70	100.00	0.2680	0.0063	**0.9686**	**0.9623**	241.4
	AER	32	45.71	0.1098	0.0072	0.9352	0.9280	210.8
	AEA	85	121.43	0.1615	**0.0013**	0.9612	0.9599	212.5
	PCA	**3**	**4.29**	**0.0780**	0.0190	0.8860	0.8669	**193.1**
DBSCAN	OF	70	100.00	0.2994	0.0119	**0.9835**	**0.9716**	11.3
	AER	10	14.29	0.0957	**0.0035**	0.9243	0.9208	7.2
	AEA	85	121.43	0.0776	0.0095	0.9234	0.9139	8.0
	PCA	**2***	**2.86***	**0.0111**	0.0043	0.9093	0.9050	**6.5**
GMM	OF	70	100.00	**0.0000** *	**0.0000** *	**0.9479**	**0.9479**	22.8
	AER	10	14.29	0.0009	0.0003	0.8850	0.8847	8.1
	AEA	85	121.43	0.0122	0.0005	0.9287	0.9282	30.9
	PCA	**2***	**2.86***	0.0002	0.0006	0.8321	0.8316	**5.7**
nuSVM	OF	70	100.00	23.1632	0.3165	**1.0000** *	0.6835	3.0
	AER	10	14.29	7.9910	0.0683	**1.0000** *	0.9317	**0.9***
	AEA	93	132.86	**2.1082**	**0.0580**	**1.0000** *	**0.9420**	1.3
	PCA	**2***	**2.86***	2.3468	0.0630	**1.0000** *	0.9370	1.0
IForest	OF	70	100.00	**4.5572**	**0.1176**	0.8893	**0.7717**	6.1
	AER	**45**	**64.29**	14.9819	0.2134	0.9192	0.7058	**5.3**
	AEA	85	121.43	16.2062	0.1836	0.9462	0.7625	**5.3**
	PCA	70	100.00	12.2452	0.2568	**0.9823**	0.7255	5.9
LOF	OF	70	100.00	0.2762	0.0028	**0.9804**	**0.9777** *	95.9
	AER	10	14.29	0.0816	0.0029	0.9157	0.9128	**2.7**
	AEA	88	125.71	0.0522	**0.0015**	0.9448	0.9432	4.5
	PCA	**3**	**4.29**	**0.0064**	0.0032	0.8838	0.8805	4.2

lowest number of output features, with an exception that arises when coupled with the IForest UML model. In this case, the transformation method that further reduces the dimensionality is the AER.

The GMM is the model that produces the least NM variance in nominal conditions (i.e. the NM produced in nominal conditions is almost constant). Concerning the other UML models, KMeans, DBSCAN and LOF experience the least variance when coupled with PCA feature transformation, while IForest produces the most stable NM when used together with original features.

Ideally, when evaluating a nominal signal, the NM should be low. The model that minimizes this value is the GMM used without any feature transformation. Considering the rest of the UML models, there is no feature transformation that appears to minimize this performance metric for most models.

On the other hand, the NM should be high when evaluating a novel signal. The nuSVM model, produces the highest NM regardless the transformations applied, but we observed that this property comes at the cost of a lower capability of generating a continuous NM value, as it tends to saturate to high values even when the signal differs only slightly from the nominal one.

The reactivity appears to be the highest when the OF are used to feed the UML models, except for the nuSVM that has the highest reactivity when combined with AEA. The analysis reveals that the LOF model with the original features shows the highest reactivity, making it the optimal model. The DBSCAN follows behind, demonstrating a similar reactivity value.

The nuSVM is the one that computes the NM the fastest. This is due to the simplicity of the inference procedure, which consists of a distance calculation. Regarding the other UML models, we observe that the fastest response happens when the dimensionality of the feature space is the lowest.

Some observations can be made about the correlation of the performance metrics. Each metric has been plotted against each other to observe emerging patterns. The two most informative plots are shown in Fig. 6. Regarding the correlation between the inference time and the number of features, the general trend suggests that the evaluation process becomes slower as the dimensionality of the feature space increases. In the second plot, the trend suggests that modifying the feature dimensionality ($FP \neq 100$) reduces the reactivity of the UML models, except for nuSVM, which exhibits an inverse

behaviour.

For completeness, the novelty metric was computed across the remaining data sets (set 1 - set 8), excluding the portion of set 1 used for training the models. The evolution of the novelty metric along all the sets for each combination is shown in Fig. 7. It appears evident that KMeans, DBSCAN, GMM and LOF provide a NM that is clearly related to the dissimilarity of the evaluated signal with respect to the nominal condition (at least related to the amplitude increase). On the other hand, as anticipated earlier, nuSVM and IForest exhibits a tendency to saturate the NM to a constant value, with an abrupt gap (almost as a binary flag). This behaviour is in contrast with the scope of our work to provide a continuously changing NM that quantifies the severity of the anomaly.

The last behaviour that emerges from Fig. 7 is that GMM is quite sensitive to the feature transformation applied. When GMM is evaluating the OF, the NM appears to be lower than when evaluating transformed features.

F. Code availability

All the functions and scripts used to obtain the results shown in this report will be available online[1].

V. Conclusion

In this work, we benchmark several unsupervised machine learning models by testing different feature transformation algorithms. We also gather a real vibration dataset to evaluate all frameworks with real-world data, including noise. The results indicate that a continuous Novelty Metric performs better with certain models — specifically KMeans, DBSCAN, GMM, and LOF — by providing an indication of novelty severity. In contrast, models like nuSVM and IF behave as a binary flag. The benchmark highlights the significance of proper feature extraction and transformation algorithms in changing the behaviour of unsupervised models. Additionally, we compute each framework's inference times to assess the complexity of the frameworks in terms of time consumption. Future work will aim to deploy and test each framework on embedded devices to provide an EDGE benchmark. We also plan to test other feature extraction algorithms, such as generative models like RealNVP, to determine if better results can be achieved. Finally, we will extend the benchmark tests to other industrial datasets to further investigate the presented study.

Acknowledgment

This work has been developed with the contribution of Politecnico di Torino Interdepartmental Centre for Service Robotics PIC4SeR[2].

[1]https://github.com/PIC4SeR/Unsupervised-Novelty-Detection-Methods-Benchmarking-with-Wavelet-Decomposition.git

[2]www.pic4ser.polito.it

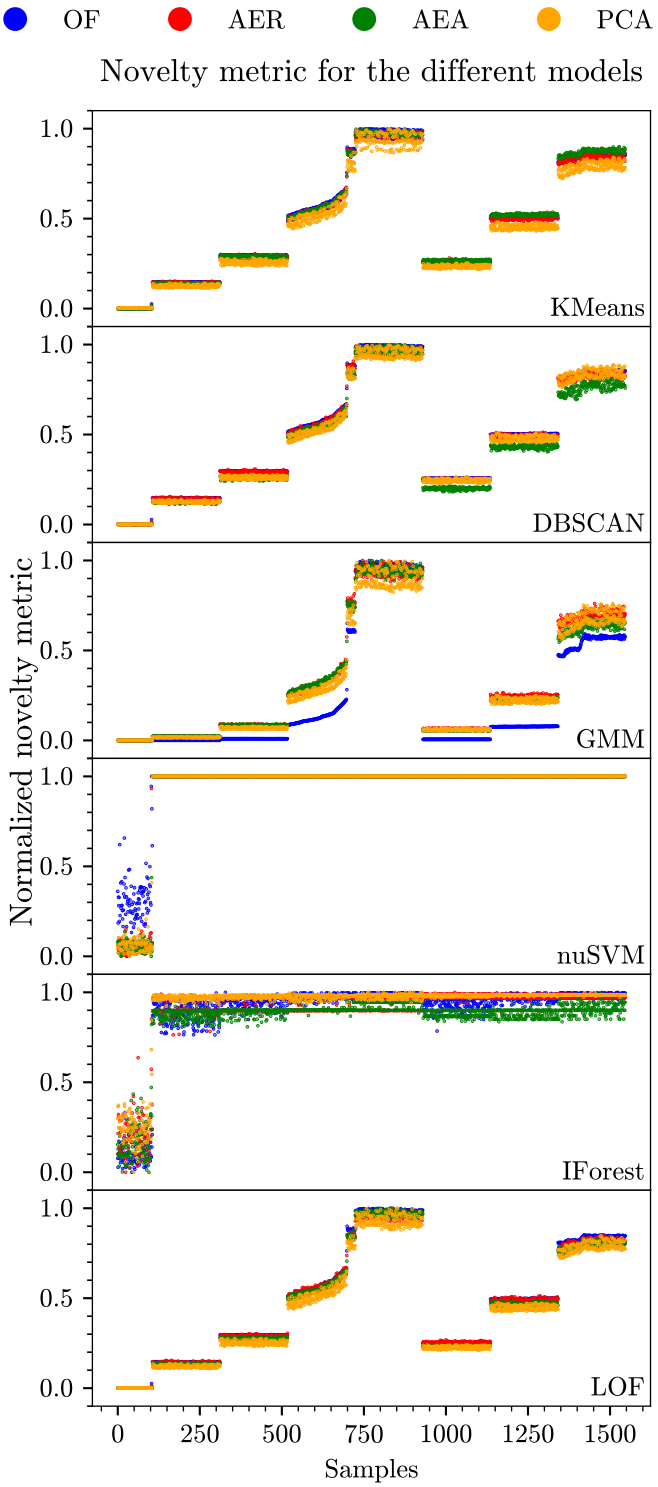

Fig. 7. Normalized novelty metric behaviour for all the combinations of UML models and feature transformation technique. In the plots, the steps (discontinuities) in the NM represent the instants in which the framework starts to evaluate a different set of data, with different characteristics. KMeans, DBSCAN and LOF appear less affected by the different feature transformation techniques if compared with the other UML models. The nuSVM novelty metric evolution shows a clear saturation for all the samples representative of novel behaviours. Moreover, it fails to produce a steady NM when evaluating samples of known modes. The Iforest model experiences almost the same saturation as the nuSVM. IForest appears also sensitive to the feature transformation used.

References

[1] Umberto Albertin et al. "A Real-Time Novelty Recognition Framework Based on Machine Learning for Fault Detection". In: *Algorithms* 16.2 (2023).

[2] Adrián Vega et al. "Malicious traffic detection on sampled network flow data with novelty-detection-based models". In: *Scientific Reports* 13 (Sept. 2023).

[3] Jeffrey Schein et al. "A rule-based fault detection method for air handling units". In: *Energy and buildings* 38.12 (2006), pp. 1485–1492.

[4] Rolf Isermann. "Model-based fault-detection and diagnosis–status and applications". In: *Annual Reviews in control* 29.1 (2005), pp. 71–85.

[5] Jihoon Chung, Bo Shen, and Zhenyu James Kong. "Anomaly detection in additive manufacturing processes using supervised classification with imbalanced sensor data based on generative adversarial network". In: *J. Intell. Manuf.* 35.5 (June 2023), pp. 2387–2406.

[6] Fa Zhu et al. "Large margin distribution multi-class supervised novelty detection". In: *Expert Systems with Applications* 224 (2023), p. 119937.

[7] Carlo Cena et al. "Physics-Informed Real NVP for Satellite Power System Fault Detection". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2024. Forthcoming.

[8] Umberto Albertin et al. *Semi-Supervised Novelty Detection for Precise Ultra-Wideband Error Signal Prediction*. 2024. arXiv: 2404.05351.

[9] Mohammad Masud et al. "Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints". In: *IEEE Transactions on Knowledge and Data Engineering* 23.6 (2011), pp. 859–874.

[10] Mohammed Chalouli, Nasr-eddine Berrached, and Mouloud Denai. "Intelligent Health Monitoring of Machine Bearings Based on Feature Extraction". In: *Journal of Failure Analysis and Prevention* 17.5 (Oct. 2017), pp. 1053–1066.

[11] Yuyan Zhang et al. "A new subset based deep feature learning method for intelligent fault diagnosis of bearing". In: *Expert Systems with Applications* 110 (2018), pp. 125–142.

[12] Qicai Zhou et al. "Degradation State Recognition of Rolling Bearing Based on K-Means and CNN Algorithm". In: *Shock and Vibration* 2019.1 (2019), p. 8471732.

[13] Luis A. Pinedo-Sánchez, Diego A. Mercado-Ravell, and Carlos A. Carballo-Monsivais. "Vibration analysis in bearings for failure prevention using CNN". In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 42.12 (Nov. 2020), p. 628.

[14] Cecilia Gattino et al. "Application of AI failure identification techniques in condition monitoring using wavelet analysis". In: *The International Journal of Advanced Manufacturing Technology* 125.9 (Apr. 2023), pp. 4013–4026.

[15] J. Lee et al. *Bearing Data Set*. IMS, University of Cincinnati, NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA. 2007.

[16] Case Western Reserve University. *Bearing Data Center: Seeded Fault Test Data*. http://data-acquisition.case.edu/. Accessed: 2024-06-05. Cleveland, OH.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012.

[18] Lanjun Wan et al. "A Novel Bearing Fault Diagnosis Method Using Spark-Based Parallel ACO-K-Means Clustering Algorithm". In: *IEEE Access* 9 (2021), pp. 28753–28768.

[19] Chen Lu et al. "Fault Diagnosis for Rotating Machinery: A Method based on Image Processing". In: *PLOS ONE* 11.10 (Oct. 2016), pp. 1–22.

[20] Jakub Spytek et al. "Novelty detection approach for the monitoring of structural vibrations using vision-based mean frequency maps". In: *Mechanical Systems and Signal Processing* 185 (2023), p. 109823.

[21] Markus Breunig et al. "LOF: Identifying Density-Based Local Outliers." In: vol. 29. June 2000, pp. 93–104.

[22] Zengyou He, Xiaofei Xu, and Shengchun Deng. "Discovering cluster-based local outliers". In: *Pattern Recognition Letters* 24.9 (2003), pp. 1641–1650.

[23] David A. Clifton, Peter R. Bannister, and Lionel Tarassenko. "Learning Shape for Jet Engine Novelty Detection". In: *Advances in Neural Networks - ISNN 2006*. Ed. by Jun Wang et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 828–835.

[24] Kemilly Dearo Garcia et al. "Online Clustering for Novelty Detection and Concept Drift in Data Streams". In: *Progress in Artificial Intelligence*. Cham: Springer International Publishing, 2019, pp. 448–459.

[25] David Arthur and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: vol. 8. Jan. 2007, pp. 1027–1035.

[26] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[27] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[28] Bernhard Schölkopf et al. "Support Vector Method for Novelty Detection". In: vol. 12. Jan. 1999, pp. 582–588.

[29] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest". In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[30] Markus M Breunig et al. "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.