machines

MDPI

Article

From Novelty Detection to a Genetic Algorithm Optimized Classification for the Diagnosis of a SCADA-Equipped Complex Machine

Luca Viale *, **Alessandro Paolo Daga**, **Alessandro Fasana** and **Luigi Garibaldi**

Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca Degli Abruzzi 24,
10129 Torino, Italy; alessandro.daga@polito.it (A.P.D.); alessandro.fasana@polito.it (A.F.);
luigi.garibaldi@polito.it (L.G.)
* Correspondence: luca.viale@polito.it

Abstract: In the field of Diagnostics, the fundamental task of detecting damage is basically a binary classification problem, which is addressed in many cases via Novelty Detection (ND): an observation is classified as novel if it differs significantly from reference, healthy data. ND is practically implemented summarizing a multivariate dataset with univariate distance information called Novelty Index. As many different approaches are possible to produce NIs, in this analysis, the possibility of implementing a simple classifier in a reduced-dimensionality space of NIs is studied. In addition to a simple decision-tree-like classification method, the process for obtaining the NIs can result as a dimension reduction method and, in turn, the NIs can be used for other classification algorithms. In addition, a case study will be analyzed thanks to the data published by the Prognostics and Health Management Europe (PHME) society, on the occasion of the Data Challenge 2021.

Keywords: machine diagnostics; novelty detection; dimension reduction method; complex machine; electronic components quality control line; SCARA robot; big data

check for updates

Citation: Viale, L.; Daga, A.P.; Fasana, A.; Garibaldi, L. From Novelty Detection to a Genetic Algorithm Optimized Classification for the Diagnosis of a SCADA-Equipped Complex Machine. *Machines* **2022**, *10*, 270. https://doi.org/10.3390/machines10040270

Academic Editors: Xiang Li and Jie Liu

Received: 8 March 2022
Accepted: 8 April 2022
Published: 9 April 2022

1. Introduction

The maintenance of a mechanical system has a fundamental role in the industrial field and has repercussions, both in terms of safety and economics, as it allows for reducing costs and downtime. In fact, in recent years, maintenance techniques have evolved rapidly, passing from corrective and preventive approaches to the most recent and developed condition-based [1] and predictive ones. Recently, research is focusing on further diagnostic techniques, aimed at prescriptive maintenance, which allows exploiting predictions to recommend operational decisions, thanks to the damage-type recognition and, consequently, its cause [2].

Among the different diagnostic techniques and prominent studies present in the literature [3–11], Novelty Detection (ND) is a classification technique based on the recognition of "abnormal" values and is frequently used for fault detection in complex industrial systems. In particular, the novelty information has a direct correspondence to the fault detection in case of exclusion of confounding influences, among which are the work and environmental conditions. ND can be based on different types of approaches, among which are distance-based and model-based approaches, support vector methods, other statistical methods, and neural networks. For example, in [12], a multivariate technique, such as Principal Component Analysis (PCA), was implemented for diagnostics via Novelty Detection. In general, ND can be seen as a classification technique between two classes (normal and abnormal or, in the context of diagnostics, healthy and damaged). From a multiclass diagnostic system point of view for prescriptive maintenance, this problem can be decomposed into multiple two-class classifications using ND, since pattern recognition usually occurs with a higher number of classes. Focusing on the monitoring of mechanical systems, ND is frequently used because it allows for recognizing the condition of machinery, when data recorded for model training is limited for abnormal classes and abundant for healthy ones. Damage simulation is, indeed, often challenging and expensive, while normal functioning is usually easy to achieve. The Novelty Indexes (NIs) exploited for ND can be obtained through different approaches, including those of a Bayesian nature, Neural Networks and Support Vector methods. In general, the effectiveness of ND techniques is evaluated both through detection rates, false alarm rates (usually summarized by the ROC curve [13]), computational costs and the complexity of the model [14,15]. For instance, Markou et al. [16] collected several neural network-based approaches for novelty detection. However, while these can be moderately effective methods, at the same time, they are complex and consequently require considerable time for generation and training. Furthermore, reduced knowledge of specific intermediate steps in neural networks is a concept called the unpredictability of Artificial Intelligence (AI) [17] and, consequently, reduces confidence in the results. For this reason, tools subject to unpredictability are not safe to use, especially when considering an AI security [18–21] and AI governance [22] context, such as the one under consideration.

This article proposes a multiclass diagnostic method, ND-based and optimized by means of a Genetic Algorithm (GA) [23]. An example of GA application is developed in [24]. The proposed method was applied to data published by PHME, which were collected on a real industrial test bench, consisting of a quality control line for electronic components. The configuration of this complex industrial line includes subsystems of different nature-rotating and non-rotating components, working in highly non-stationary conditions and with several damage and failure conditions.

The results of the proposed ND-based technique have been evaluated with different performance indices (e.g., accuracy), obtained through six different types of classifiers [25,26] (Linear Discriminant Analysis, k-Nearest Neighbors, Decision Tree, Linear Support Vector Machine, Gaussian Naive Bayes and Kernel Naive Bayes). In addition to the positive results obtained in terms of performance, the proposed work further allows both reducing the dataset dimensions (and, consequently, the amount of data to be stored and processed) and speeding up the diagnostic process, by making the fault recognition timely. Thanks to the latter property, the proposed method is particularly suitable for online applications and allows for reducing the effects of the curse of dimensionality [5,27].

Finally, it is worth noting that a generic data-to-decision (D2D) process taken from the implementation of structural health monitoring [6,28] is mainly composed of the following six phases: (1) operational evaluation, (2) data acquisition and cleansing, (3) signal processing (i.e., features selection, extraction, and metrics), (4) pattern processing (i.e., statistical model development and validation), (5) situation assessment and (6) decision making. Considering this waterfall structure, it is possible to affirm that the proposed work focuses mainly on (4), since the introduced novelty concerns data mining for damage identification. These techniques generally follow a consolidated hierarchical structure of steps, defined in [29,30]. The foremost levels of this structure can be summarized as: (a) detection (a qualitative indication of the damage presence), (b) localization (damage position), (c) classification (damage type), (d) assessment (damage size) and (e) prediction (degree of safety and remaining useful life estimation). Among the levels (a–d) concerning diagnostics, this work aims both to detect the presence of damage and to classify the damage itself, thanks to class recognition. The scientific literature contains numerous articles, in which these aspects, concerning fault diagnosis, are deepened, as in [31–34].

The article is structured as follows. Section 2 contains a description of the mechanical system adopted as case study and of the dataset used for the application of the proposed methodology, which is described in Section 3. Finally, the results and conclusions are reported in Sections 4 and 5, respectively.

2. Case Study

The proposed diagnostic method was developed for industrial applications and, in particular, was tested on the dataset that was distributed for the Prognostics and Health

Management Europe (PHME) society Data Challenge 2021 [35]. This section describes the dataset and the related test bench used for its acquisition.

2.1. Test Bench Description

The machine used by PHME is a complex system, composed primarily of the 4-axis SCARA robot shown in Figure 1. It represents a typical component for the quality control of an industrial production line. Electric fuses are tested on this test bench (electricity conduction, temperature reached by induced heating), taken with a vacuum gripper. For these controls and for real-time monitoring of the machinery health state, a Supervisory Control and Data Acquisition (SCADA) system composed of 50 sensors was implemented to record the evolution of the quantities of interest and to consider the contributions of the several different components. Please note that the main components making up the entire machine are the 4-axis SCARA robot, a thermal imaging camera and a camera for detecting fuses, Electronically Commutated (EC) and Direct-Current (DC) motors, a pneumatic system, including vacuum pumps and various valves and an electrical power supply circuit for control tests. As can be noted, the overall structure of the machine is rather complex and heterogeneous. There are components of a different nature (rotating and non-rotating parts, electric and pneumatic equipment) and this makes the extraction of the most representative features for the machinery health conditions more challenging.

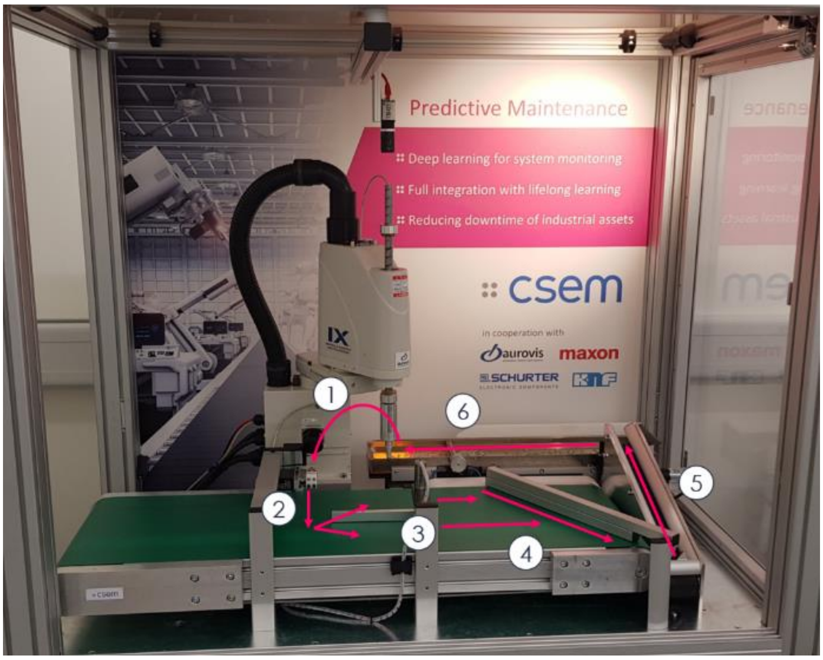
Predictive Maintenance
Deep learning for system monitoring
Full integration with lifelong learning
Reducing downtime of industrial assets
csem
in cooperation with
aurovis
maxon
SCHURTER
KNF
1
2
3
4
5
6

Figure 1. Equipment: 4-axis SCARA robot picking up electrical fuses with a vacuum gripper, from a feeder to a fuse-test-bench for large-scale quality control.

There are no defects throughout the entire quality control line during the tests carried out under healthy conditions. The five different artificial failure forms introduced were obtained by manually altering one or more components. The five introduced faults affect the sensor readings in different ways, so this dataset potentially allows one to classify not only the presence of defects but also their type, from a prescriptive maintenance point of view.

2.2. Dataset Description

The recorded experimental dataset is relative to 50 signals, inherent to different quantities of interest. These quantities vary from measurements of ambient temperature and humidity to pressure measurements inherent to the state of the machine up to quantities of a different nature, such as CPU temperature and process memory consumption. Each of these signals is described in the dataset through a specific set of fields, inherent to a fixed time window ($vCnt$ = number of samples recorded; $vFreq$ = Sampling frequency; $vMax$ = Maximum recorded value; $vMin$ = Minimum recorded value; $vTrend$ = trend of the historical series and $value$ = Average value) to describe its characteristics. Appendix A lists the different signals present in the dataset and the related measured fields per sensor. The reference time windows have a duration of 10 s, while each experiment can last from 1 to 3 h approximately. However, the dataset was pre-processed by averaging the available features over the entire acquisition period (i.e., $\approx$ 1 to 3 h) to limit the dataset and to obtain unique features describing each experiment. Furthermore, not all measures have all the above characteristics. Therefore, the dimension of the data matrix X_0 of dimensions $m \times n$ is composed of $m = 70$ rows for $n = 240$ columns (where m represents the tests and n the features). Further, 50 tests were performed under healthy conditions, while the five conditions with different failures have a cardinality of 4 tests each, for a total of 70 tests.

It should be noted that, in the context of diagnostics and health monitoring of mechanical systems, data are generally collected using suitable sensors (e.g., accelerometers, load cells and temperature sensors) positioned on the machinery of interest, both during its operation in optimal conditions and with the presence of (alternatively simulating) faults, defects, damage, failures. Therefore, each performed test is classified through a specific label describing the condition of the machinery. In the following, healthy conditions of the machinery will be indicated as *Class* 0, while the damage will be generally defined as *Class* k, where $k \in \mathbb{N}$ is the number of types of damage considered.

Finally, this dataset was further pre-processed by standardizing the data on the mean value and on the standard deviation, referred to as the healthy class, obtaining the X matrix of size $m \times n$ and rank $L \leq min(m, n)$. Vector C, containing the labels, describes the condition of the machinery concerning each test carried out and, consequently, has dimensions of $1 \times m$.

3. Proposed Methodology

Novelty Detection (ND) is a semi-supervised methodology for implementing a binary classification problem using only data from a healthy reference condition. When a new datapoint arrives, its distance from the healthy reference data cloud is measured, and this measure, usually called Novelty Index (NI), is compared to a threshold so as to identify if the new data are sufficiently far away to be considered *novel*. The NI is then a reduced dimensionality (1-D) version of the original multivariate dataset.

Different algorithms are available for computing NIs. The simplest involves the projection of the multivariate dataset along a direction which is believed to correspond to the damage-evolution direction. In this case, the *NI* is simply a linear combination of features (i.e., a weighted sum of features). Hence, the training consists in the determination of such weights α.

$$NI_{\alpha} = X \cdot \alpha \tag{1}$$

In this work this task has been tackled by a heuristic maximization of a utility function measuring the separation of the different classes along the direction identified by α.

A genetic algorithm was used to find the optimal α minimizing the p-value from an ANOVA post-hoc test indicating the degree of separation. In particular, the average value of the p-values which represents the degree of separation between each pair of classes has been optimized. In this way, not only the ability of features to identify classes was considered but the number of distinguished classes was also maximized.

To improve the identification of classes, a more refined *NI* was implemented, based on Mahalanobis Distance (*MD*) [36]:

$$NI_m = MD^2 = X' \Sigma^{-1} X \tag{2}$$

where Σ is the covariance matrix of the reference class. Considering that datasets with a high number of features are usually handled, it may often be impossible to correctly estimate Σ^{-1} using a few healthy points in a feature space with a large dimension.

An optimization scheme similar to the previously introduced one, based on GA, was then proposed for the selection of a subspace with a lower dimension. GA was then iterated several times changing the dimensionality to find the optimal features to be kept for the computation of NI_m. Comparing the optimal utility function values, it is possible to select the final subspace. Appendix A shows the results of the performed GA optimization in terms of coefficients referred to each feature which allow obtaining NI_α and NI_m.

Merging the information of NI_α and NI_m in a 2-D space, the classification task can be implemented with better results. This proves that the implementation of a simple classifier in a 2D space of NIs (such as a decision tree that basically allows recognizing the damage class by dividing the resulting 2D space into regions) can be used for multi-class classification purposes. In addition to this possible use, NIs are also suitable for the implementation of other classifiers, as shown in Section 4. In general, the proposed method can be considered as a size reduction method for classification algorithms.

To conclude, a flowchart is presented in Figure 2 to summarize and clarify the proposed method.

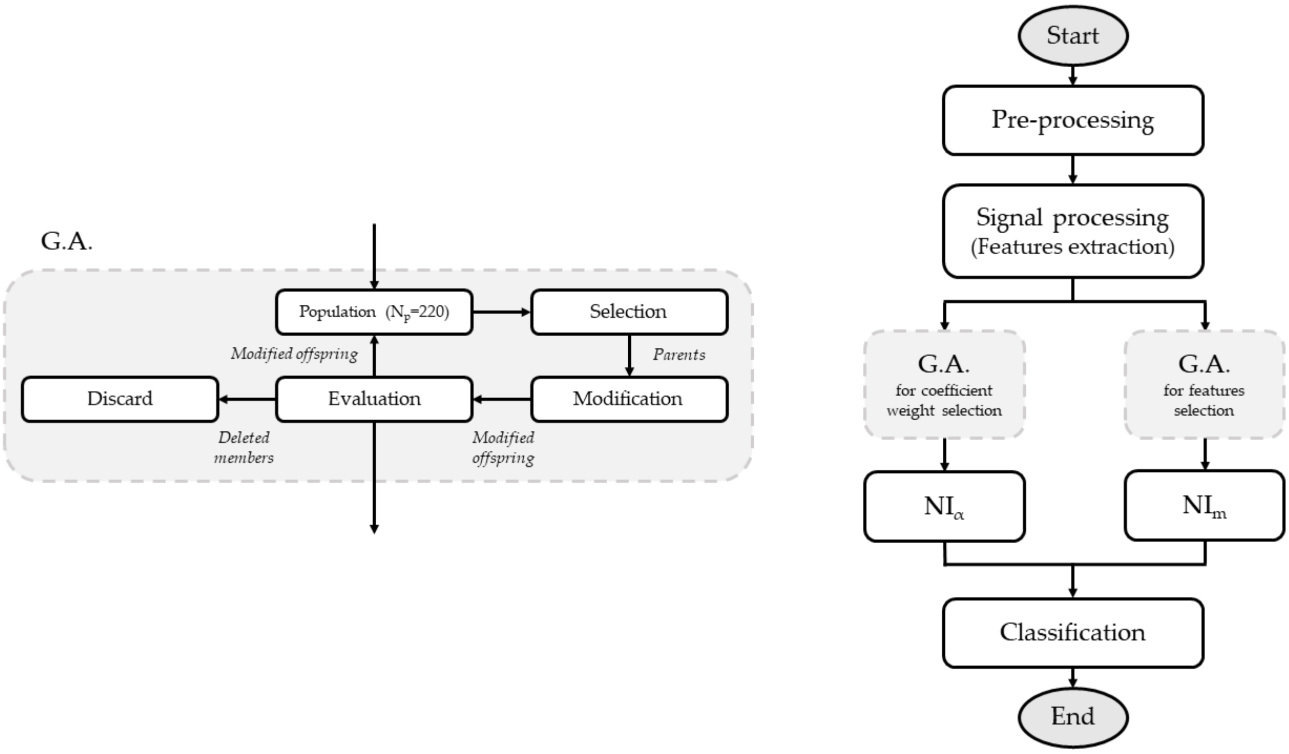

Figure 2. Flowchart summarizing the proposed method with a focus on the GA cycle.

4. Results and Discussion

This section shows the results obtained with the proposed method applied on the database described in Section 2. After having obtained the NIs in a reduced-dimension space, six of the main classification models were applied: Linear Discriminant Analysis (LDA) [37], k-Nearest Neighbor (kNN) with $k = 2$, given the small amount of data for the minority classes [38], Decision Trees (DT) [39], Linear Support Vector Machine (SVM) [40], Gaussian Naive Bayes (GNB) and Kernel Naive Bayes (KNB) [41]. These classifiers have been adopted both because they are the most widely used (semi-)supervised machine learning algorithms, and to study the proposed method performances as the type of algorithm varies.

A Monte Carlo Cross-Validation (MCCV) [42] was applied to all the tests to obtain more precise results, in terms of performance indices. Indeed, since a classification model needs both a training dataset and a second group of data for verification (called *validation dataset*), the choice of these datasets can take place in different ways. A k-fold Cross-Validation (CV) consists of dividing data into k groups. Only one group is used as a validation dataset, while the remaining $k-1$ as training. This process is repeated k times until all groups have been validated. In this case, given that the number of samples describing classes with damage is reduced to four examples per class, $k = 4$ was chosen to have at least one test in each subdivision of the dataset and to train the model correctly. Considering a generic CV on a database composed of n samples, divided, respectively, into n_t for the training set and $n_v = n - n_t$ for the validation set, the binomial coefficient $\begin{pmatrix} n \\ n_v \end{pmatrix}$ represents the number of different combinations for the subdivision. However, each of these subdivisions can bring different results, in terms of model generation and, consequently, accuracy. MCCV is a very effective method that consists, in addition to the random subdivision of the samples into the training and validation groups, of the iteration of this procedure $N = 50$ times. Thanks to this method, the computational complexity is significantly reduced, and the average accuracy tends to the theoretical value of the generated model. Figure 3 shows an example of accuracy trends calculated with MCCV as the number of iterations N increases to demonstrate their convergence.

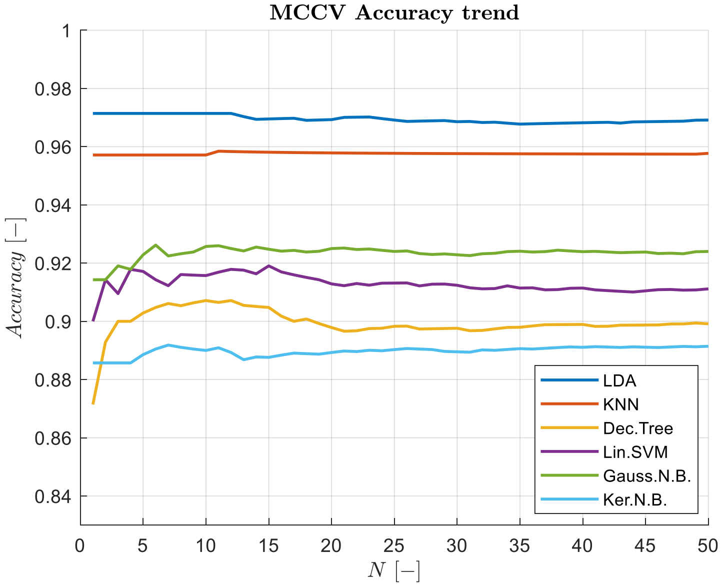

Figure 3. The trend of the accuracy values calculated with MCCV on the merged information of NI_α and NI_m as the number of iterations N grows.

The proposed method performance will be evaluated through different comparative indices [43]. In addition to the typical accuracy, other indices are used in this study compared to traditional sensitivity and specificity, since the database used is multi-class and the proposed method aims to recognize not only the damage but also its nature. For this reason, considering a generic confusion matrix, as in Table 1, the following indices are introduced to evaluate the performance of the methods, where the acronyms are as in Table 1 ($TN = TN_0$, $FP = \sum_{k=1}^{K} FP_k$, $FN = \sum_{k=1}^{K} FN_k$, $TP = \sum_{k=1}^{K} TP_k$, $CE = \sum_{k=1}^{K}\sum_{k=1}^{K} CE_{k,k}$, $TC = TP + TN + FP + FN + CE$):

- **Accuracy**: this represents the ability of the classifier to correctly recognize positive and negative cases.

$$Acc = \frac{TP + TN}{TC} \tag{3}$$

- **Missed Alarms:** this indicates cases where degradation exists, but the classifier cannot recognize it.

$$M.A. = \frac{FN}{TC} \tag{4}$$

- **False Alarms**: unlike Missed Alarms, this represents the percentage of cases out of the total in which the machinery is healthy, but the algorithm assumes damage. Both the F.A.s and the M.A.s are important indices and could be preferred over the accuracy for safety or economic reasons.

$$F.A. = \frac{FP}{TC} \tag{5}$$

- **Class Errors Rate**: this index allows for recognizing how many tests have not been correctly classified, despite being recognized as unhealthy. Therefore, it represents the error made in identifying the specific damage.

$$C.E.\ R. = \frac{CE}{TC} \tag{6}$$

- **Performance Index**: this is a redundant index, as it is the product of the indices seen so far, but allows for observing, simultaneously, the set of previous performances.

$$P.I. = Acc \times (1 - M.A.) \times (1 - F.A.) \times (1 - C.E.) \tag{7}$$

- **Frobenius Norm**: this is a matrix norm defined as the square root of the sum of the absolute squares of its elements.

$$\parallel A \parallel_F = \sqrt{\sum_{i=1}^{k+1} \sum_{j=1}^{k+1} |a_{ij}|^2} \tag{8}$$

  where A is the confusion matrix after having standardized it by columns and subtracting the identity matrix, a_{ij} are the elements of the matrix A and k are the numbers of the fault classes. In this way, the results obtained in terms of the Frobenius norm will be greater than or equal to 0. In detail, the larger the norm, the worse the classification and vice versa.
- **AUC**: the area under the receiver operating characteristic (ROC) curve. The AUC provides a combined measure of performance across all possible classification thresholds.

Table 1. A generic confusion matrix (TP = True Positive, FP = False Positive, FN = False Negative, TN = True Negative, CE = Class Error, TC = Total Cases) where $k = 0$ is the reference class and $k = 1, \ldots, K$ are the considered damage classes.

	True Class 0	**True Class 1**	**True Class . . .**	**True Class k**	**True Class . . .**	**True Class K**
Predicted Class 0	TN_0	FN_1	$\mathrm{FN}_{\ldots}$	FN_k	$\mathrm{FN}_{\ldots}$	FN_K
Predicted Class 1	FP_1	TP_1	$\mathrm{CE}_{1,\ldots}$	$\mathrm{CE}_{1,k}$	$\mathrm{CE}_{1,\ldots}$	$\mathrm{CE}_{1,K}$
Predicted Class . . .	$\mathrm{FP}_{\ldots}$	$\mathrm{CE}_{\ldots,1}$	$\mathrm{TP}_{\ldots}$	$\mathrm{CE}_{\ldots,k}$	CE	CE
Predicted Class k	FP_k	$\mathrm{CE}_{k,1}$	$\mathrm{CE}_{k,\ldots}$	TP_k	$\mathrm{CE}_{k,\ldots}$	$\mathrm{CE}_{k,K}$
Predicted Class . . .	$\mathrm{FP}_{\ldots}$	$\mathrm{CE}_{\ldots,1}$	$\mathrm{CE}_{\ldots,\ldots}$	$\mathrm{CE}_{\ldots,k}$	$\mathrm{TP}_{\ldots}$	$\mathrm{CE}_{\ldots,K}$
Predicted Class K	FP_K	$\mathrm{CE}_{K,1}$	$\mathrm{CE}_{K,\ldots}$	$\mathrm{CE}_{K,k}$	$\mathrm{CE}_{K,\ldots}$	TP_K

Among the many different algorithms for calculating NIs, this method initially provides the projection of the multivariate dataset along a direction that is believed to correspond to the evolution of the damage. A Genetic Algorithm was adopted to optimize the results and, thus, to maximize the number of distinct classes. The results concerning NI_α are shown in Figure 4. As is clear from the image, classes 2 and 3 definitely stand out from the healthy values; nevertheless, classes 5, 7 and 9 are more difficult to identify.

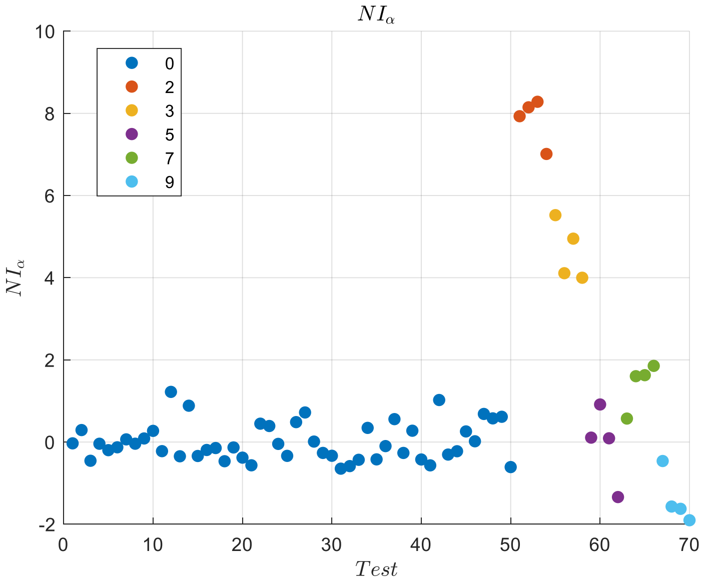

Figure 4. Linear Combination NI_α. Healthy conditions of the machinery are indicated as Class 0, while the damage is defined as Class k, where k = 2, 3, 5, 7, 9.

Because of this, a more refined NI was calculated, based on Mahalanobis Distance (MD). The resulting MD-NIs from such a subspace are plotted in Figure 5. As can be noticed, classes 2 and 3 definitely stand out again from the healthy values, but classes 5 and 7 are now better identifiable. In any case, a perfect classification is still impossible.

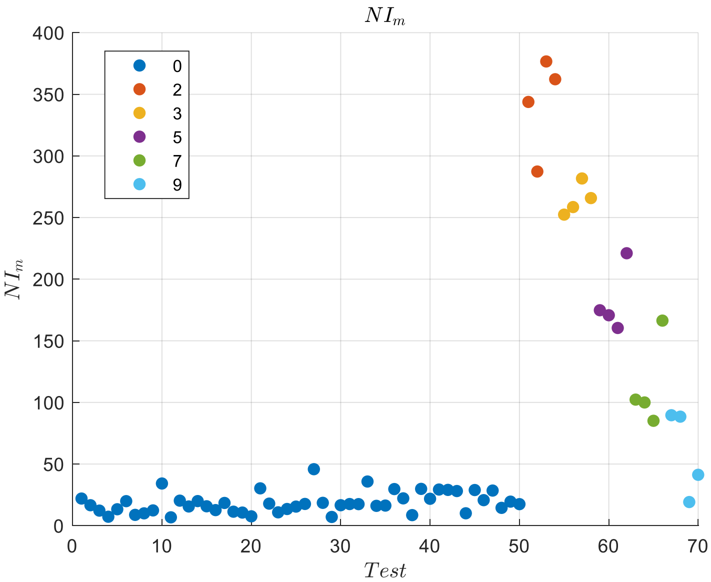

Figure 5. Mahalanobis Distance NI_m. Healthy conditions of the machinery are indicated as Class 0, while the damage is defined as Class k, where k = 2, 3, 5, 7, 9.

However, merging the information of NI_α and NI_m in a 2-D space (Figure 6), the classification task can be implemented with better results. In this particular case, the classifier was built by segmenting the 2-D space in rectangular regions, as visible in Figure 6. After reducing the original space (having 240 dimensions) to a 2-D space (where the two dimensions correspond to the calculated NIs), it is possible to calculate the performance indices obtained with each classifier and initially compare them with those detectable using the initial features. These performance indices are shown in Tables 2 and 3.

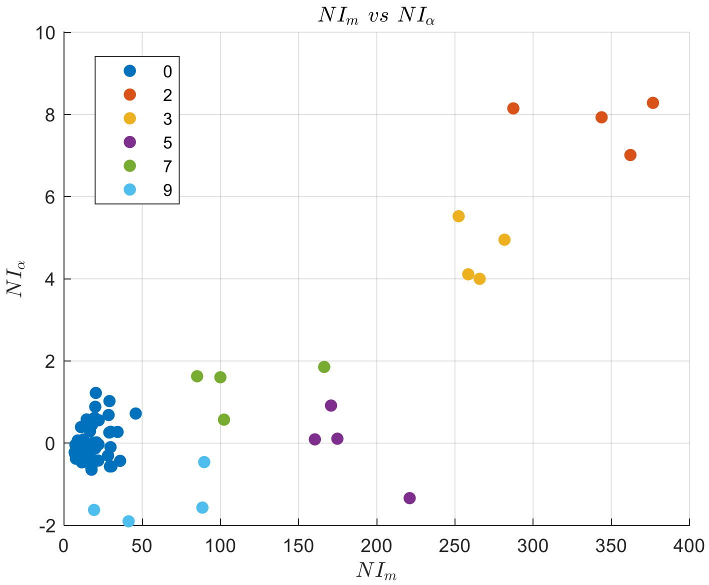

Figure 6. Mahalanobis Distance NI_m vs. Linear Combination NI_α. Healthy conditions of the machinery are indicated as Class 0, while the damage is defined as Class k, where k = 2, 3, 5, 7, 9.

Table 2. Comparison indices obtained with the features in the original space.

Index	LDA	KNN	Decision Tree	Linear SVM	Gaussian N.B.	Kernel N.B.
Accuracy	-	81.6%	75.7%	55.9%	-	71.4%
Missed Alarms	-	15.6%	6.3%	10.3%	-	28.6%
False Alarms	-	1.4%	8.0%	23.4%	-	0.0%
Class Errors	-	1.4%	10.0%	10.4%	-	0.0%
P.I.	-	67.0%	58.7%	34.4%	-	51.0%
Frobenius N.	-	2.35	2.05	2.16	-	3.16
AUC	-	0.99	1.00	0.80	-	1.00

Table 3. Comparison indices obtained with the features calculated thanks to the Multi-ND method optimized by means of GA.

Index	LDA	KNN	Decision Tree	Linear SVM	Gaussian N.B.	Kernel N.B.
Accuracy	96.9%	95.8%	89.9%	91.1%	92.4%	89.1%
Missed Alarms	1.6%	1.9%	0.3%	1.7%	0.4%	3.7%
False Alarms	0.0%	0.0%	1.0%	0.3%	1.0%	0.5%
Class Errors	1.5%	2.3%	8.9%	6.9%	6.2%	6.7%
P.I.	93.9%	91.8%	80.9%	83.1%	85.5%	79.7%
Frobenius N.	0.54	0.73	1.60	0.98	0.92	1.33
AUC	1.00	1.00	1.00	1.00	1.00	1.00

In Table 4 it is possible to note that all the performance indices concerning the model precision are significantly improved thanks to the proposed method. On the other hand, the indices indicating the different error typologies decrease on average.

Table 4. Percentage variation of the performance indices obtained with the proposed method and with classifications using the features extracted from the original space. Frobenius Norms and AUC percentages are standardized. Please note that all the percentage variations represent an improvement in performance since the percentage values referring to accuracy, P.I. and AUC indicate an increase in precision, while those referring to Missed Alarms, False Alarms, Class Errors and Frobenius Norms indicate a decrease in errors.

Index	KNN	Decision Tree	Linear SVM	Kernel N.B.
Accuracy	14.2%	14.3%	35.2%	17.7%
Missed Alarms	13.7%	6.1%	8.6%	24.9%
False Alarms	1.4%	7.0%	23.1%	-0.5%
Class Errors	−0.9%	1.2%	3.5%	-6.7%
P.I.	24.8%	22.3%	48.7%	28.7%
Frobenius N.	221.5%	28.1%	121.4%	137.8%
AUC	0.6%	0.2%	20.1%	0.0%

In addition, it can be noted that the variations relating to LDA and GNB classifiers are not present, since it is not possible to use them with the features extracted from the original space. In fact, given that the proposed method allows for reducing the space dimensionality, it makes it possible to employ classifiers otherwise not usable.

To conclude, in Table 5, it can be further observed how the classification operation with the use of NIs is significantly speeded up. In particular, the proposed method allows for reducing the computational effort by 97% (reducing the average elapsed time per cycle from 17.40 s, using the original dataset, to 0.57 s, employing the NIs obtained thanks to the proposed method). The reported results were obtained by averaging the time taken over 50 cycles. The computational software used to conduct these experiments is MATLAB R2020b, running on a PC equipped with a 10th gen Intel i7 processor and 16 GB RAM.

Table 5. Processing times of the classification with different reduced datasets.

Dataset	Average Elapsed Time (%) per Cycle
Original dataset with "n" features	100.0%
Multi − NI optimized by means of GA (NI_α, NI_m)	3.3%

5. Conclusions

This work exploits simple novelty detection strategies to produce a 2-D space where a classification is possible, in an easy but satisfactory way. The proposed method was described and subsequently applied to a real industrial case, consisting of a complex quality control line of electronic components. In particular, the first axis is obtained as a linear combination of original features. The second axis is obtained as the (Mahalanobis) distance of a new data point from a reference distribution in a subspace composed of 19 selected features. Since it is a parametric model, both such features and the linear combination weights were automatically selected by a routine able to optimize a measure of class separation by means of a genetic algorithm. This composition of the features made it possible to extract the most relevant information in relation to the machinery state of health. Despite the presence of components heterogeneous in nature and non-stationary working conditions, the results seem to suggest that such a 2-D data compression can lead to satisfactory diagnostic results, improving the performance of a simple feature extraction technique. In particular, the results showed an improvement in terms of the general performance index, ranging from 22% to 49% in relation to the classification algorithm.

In addition to this advantage, the proposed method is able to recognize not only the failure condition of the mechanical system (damage detection) but also the type of damage (damage classification). This characteristic makes the method suitable for a prescriptive maintenance conception.

In general, in addition to being a classification ND-based method, the proposed work can also be applied as a dimension reduction method, since it allows for improving the diagnostic results by simultaneously and significantly decreasing the number of features. This is very important when dealing with big data [44]. This aspect has further related advantages, such as the memory reduction for saving data for diagnostic purposes and the speed increase in the calculation of the predictions. Indeed, it was possible to observe a reduction of about 97% of calculation time compared to the classification with the original features dataset. This last advantage makes the method suitable for real-time applications or for applications where timely damage recognition is particularly essential.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The employed dataset was provided online at https://github.com/PHME-Datachallenge/Data-Challenge-2021 on the occasion of the PHME Data Challenge 2021. Accessed 22 March 2022.

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

Appendix A

Table A1 shows the list of the signals present in the PHME dataset and the fields measured per sensor. Furthermore, Figure A1 shows α indices and the features selected by means of GA among those considered for the analysis.

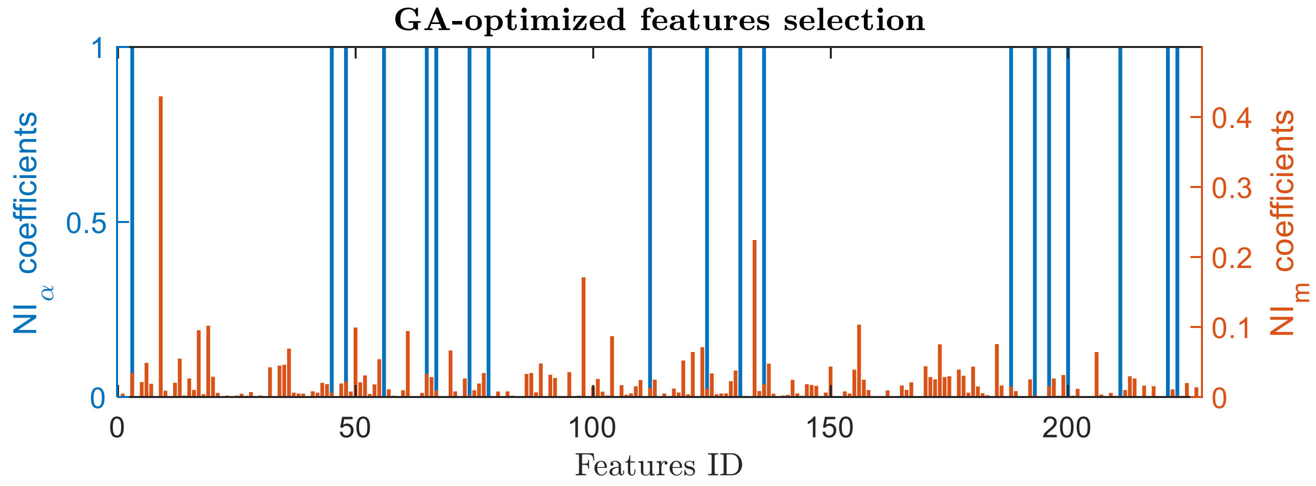

Figure A1. Coefficient values obtained thanks to GA optimization for the calculation of NI_α (in blue) and NI_m (in red).

Table A1. List of the signals present in the PHME dataset and the related fields measured per sensor.

Sensors	vCnt	vFreq	vMax	vMin	vStd	vTrend	Value
CpuTemperature			X	X	X		X
DurationPickToPick	X	X	X	X	X	X	X
DurationRobotFromFeederToTestBench	X	X	X	X	X	X	X
DurationRobotFromTestBenchToFeeder	X	X	X	X	X	X	X
DurationTestBenchClosed	X	X	X	X	X	X	X
EPOSCurrent	X	X	X	X	X	X	X
EPOSPosition	X	X	X	X	X	X	X
EPOSVelocity	X	X	X	X	X	X	X
ErrorFrame	X	X					
FeederAction1	X						
FeederAction2	X						
FeederAction3	X						
FeederAction4	X						
FeederBackgroundIlluminationIntensity	X	X	X	X	X	X	X
FuseCycleDuration	X	X	X	X	X	X	X
FuseHeatSlope	X	X	X	X	X	X	X
FuseHeatSlopeNOK	X	X	X	X	X	X	X
FuseHeatSlopeOK	X	X	X	X	X	X	X
FuseIntoFeeder	X						
FuseOutsideOperationalSpace	X	X	X	X	X	X	X
FusePicked	X	X	X	X	X	X	X
FuseTestResult	X	X	X	X	X	X	X
Humidity							X
IntensityTotalImage	X	X	X	X	X	X	X
IntensityTotalThermoImage	X	X	X	X	X	X	X
LightBarrieActiveTaskDuration2	X	X					
LightBarrierActiveTaskDuration1	X	X	X	X	X	X	X
LightBarrierActiveTaskDuration1b	X	X					
LightBarrierPassiveTaskDuration1	X	X	X	X	X	X	X
LightBarrierPassiveTaskDuration1b	X	X					
LightBarrierPassiveTaskDuration2	X	X					
LightBarrierTaskDuration	X	X					
NumberEmptyFeeder	X						
NumberFuseDetected	X	X	X	X	X	X	X
NumberFuseEstimated	X	X	X	X	X	X	X
Pressure	X	X	X	X	X	X	X
ProcessCpuLoadNormalized			X	X	X		X
ProcessMemoryConsumption			X	X	X		X
SharpnessImage	X	X	X	X	X	X	X
SmartMotorPositionError	X	X	X	X	X	X	X
SmartMotorSpeed	X	X	X	X	X	X	X
Temperature							X
TemperatureThermoCam	X	X	X	X	X	X	X
TotalCpuLoadNormalized			X	X	X		X
TotalMemoryConsumption			X	X	X		X
Vacuum	X	X	X	X	X	X	X
VacuumFusePicked	X	X	X	X	X	X	X
VacuumValveClosed	X	X	X	X	X	X	X
ValidFrame	X	X					
ValidFrameOptrisPIIRCamera	X	X					

References

1. Jardine, A.K.S.; Lin, D.; Banjevic, D. A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance. *Mech. Syst. Signal Processing* **2006**, *20*, 1483–1510. [CrossRef]
2. Gordon, C.A.K.; Burnak, B.; Onel, M.; Pistikopoulos, E.N. Data-Driven Prescriptive Maintenance: Failure Prediction Using Ensemble Support Vector Classification for Optimal Process and Maintenance Scheduling. *Ind. Eng. Chem. Res.* **2020**, *59*, 19607–19622. [CrossRef]

3. Daga, A.P.; Garibaldi, L.; Fasana, A.; Marchesiello, S. ANOVA and Other Statistical Tools for Bearing Damage Detection. In Proceedings of the International Conference Surveillance, Fez, Morocco, 23 May 2017; pp. 22–24.
4. Castellani, F.; Garibaldi, L.; Daga, A.P.; Astolfi, D.; Natili, F. Diagnosis of Faulty Wind Turbine Bearings Using Tower Vibration Measurements. *Energies* **2020**, *13*, 1474. [CrossRef]
5. Daga, A.P.; Fasana, A.; Marchesiello, S.; Garibaldi, L. The Politecnico Di Torino Rolling Bearing Test Rig: Description and Analysis of Open Access Data. *Mech. Syst. Signal Processing* **2019**, *120*, 252–273. [CrossRef]
6. Daga, A.P.; Garibaldi, L. Machine Vibration Monitoring for Diagnostics through Hypothesis Testing. *Information* **2019**, *10*, 204. [CrossRef]
7. Natili, F.; Daga, A.P.; Castellani, F.; Garibaldi, L. Multi-Scale Wind Turbine Bearings Supervision Techniques Using Industrial SCADA and Vibration Data. *Appl. Sci.* **2021**, *11*, 6785. [CrossRef]
8. Yan, A.-M.; Kerschen, G.; De Boe, P.; Golinval, J.-C. Structural Damage Diagnosis under Varying Environmental Conditions—Part I: A Linear Analysis. *Mech. Syst. Signal Processing* **2005**, *19*, 847–864. [CrossRef]
9. Worden, K. Structural Fault Detection Using a Novelty Measure. *J. Sound Vib.* **1997**, *201*, 85–101. [CrossRef]
10. Worden, K.; Manson, G.; Fieller, N.R.J. Damage Detection Using Outlier Analysis. *J. Sound Vib.* **2000**, *229*, 647–667. [CrossRef]
11. Bull, L.A.; Worden, K.; Fuentes, R.; Manson, G.; Cross, E.J.; Dervilis, N. Outlier Ensembles: A Robust Method for Damage Detection and Unsupervised Feature Extraction from High-Dimensional Data. *J. Sound Vib.* **2019**, *453*, 126–150. [CrossRef]
12. Daga, A.P.; Fasana, A.; Garibaldi, L.; Marchesiello, S. On the Use of PCA for Diagnostics via Novelty Detection: Interpretation, Practical Application Notes and Recommendation for Use. In Proceedings of the PHM Society European Conference, Turin, Italy, 1 July 2020; Volume 5, p. 13.
13. Marzban, C. The ROC Curve and the Area under It as Performance Measures. *Weather. Forecast.* **2004**, *19*, 1106–1114. [CrossRef]
14. Pimentel, M.A.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A Review of Novelty Detection. *Signal Processing* **2014**, *99*, 215–249. [CrossRef]
15. Japkowicz, N.; Myers, C.; Gluck, M. A Novelty Detection Approach to Classification. In Proceedings of the Fourteenth Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 20–25 August 1995; pp. 518–523.
16. Markou, M.; Singh, S. Novelty Detection: A Review—Part 2: Neural Network Based Approaches. *Signal Processing* **2003**, *83*, 2499–2521. [CrossRef]
17. Yampolskiy, R.V. Unpredictability of AI: On the Impossibility of Accurately Predicting All Actions of a Smarter Agent. *J. Artif. Intell. Conscious.* **2020**, *7*, 109–118. [CrossRef]
18. Babcock, J.; Kramár, J.; Yampolskiy, R.V. Guidelines for Artificial Intelligence Containment. *Next-Gener. Ethics Eng. A Better Soc.* **2019**, 90–112.
19. Trazzi, M.; Yampolskiy, R.V. Building Safer AGI by Introducing Artificial Stupidity. *arXiv* **2018**, arXiv:1808.03644. [CrossRef]
20. Behzadan, V.; Munir, A.; Yampolskiy, R.V. A Psychopathological Approach to Safety Engineering in Ai and Agi. In Proceedings of the International Conference on Computer Safety, Reliability, and Security, Västerås, Sweden, 18–21 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 513–520. [CrossRef]
21. Ozlati, S.; Yampolskiy, R. The Formalization of Ai Risk Management and Safety Standards. In Proceedings of the Workshops at the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA; 2017.
22. Ramamoorthy, A.; Yampolskiy, R. Beyond Mad? The Race for Artificial General Intelligence. *ITU J.* **2018**, *1*, 77–84.
23. Whitley, D. A Genetic Algorithm Tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [CrossRef]
24. Daga, A.P.; Garibaldi, L. GA-Adaptive Template Matching for Offline Shape Motion Tracking Based on Edge Detection: IAS Estimation from the SURVISHNO 2019 Challenge Video for Machine Diagnostics Purposes. *Algorithms* **2020**, *13*, 33. [CrossRef]
25. Sammut, C.; Webb, G.I. *Encyclopedia of Machine Learning*; Springer Science & Business Media: Boston, MA, USA, 2011.
26. MacKay, D.J.; Mac Kay, D.J. *Information Theory, Inference and Learning Algorithms*; Cambridge University Press: Cambridge, UK, 2003.
27. Köppen, M. The Curse of Dimensionality. In Proceedings of the 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), On the Internet (World-Wide-Web), 4–18 September 2000; Volume 1, pp. 4–8.
28. Farrar, C.R.; Doebling, S.W. Damage Detection and Evaluation II. In *Modal Analysis and Testing*; Silva, J.M.M., Maia, N.M.M., Eds.; Springer: Dordrecht, The Netherlands, 1999; pp. 345–378. ISBN 978-94-011-4503-9.
29. Rytter, A. *Vibrational Based Inspection of Civil Engineering Structures*; Fracture and Dynamics; Dept. of Building Technology and Structural Engineering, Aalborg University: Aalborg, Denmark, 1993.
30. Worden, K.; Dulieu-Barton, J.M. An Overview of Intelligent Fault Detection in Systems and Structures. *Struct. Health Monit.* **2004**, *3*, 85–98. [CrossRef]
31. Gao, Z.; Cecati, C.; Ding, S.X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [CrossRef]
32. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of Machine Learning to Machine Fault Diagnosis: A Review and Roadmap. *Mech. Syst. Signal Processing* **2020**, *138*, 106587. [CrossRef]
33. Liu, R.; Yang, B.; Zio, E.; Chen, X. Artificial Intelligence for Fault Diagnosis of Rotating Machinery: A Review. *Mech. Syst. Signal Processing* **2018**, *108*, 33–47. [CrossRef]

34. Lo, N.G.; Flaus, J.-M.; Adrot, O. Review of Machine Learning Approaches in Fault Diagnosis Applied to IoT Systems. In Proceedings of the 2019 International Conference on Control, Automation and Diagnosis (ICCAD), Grenoble, France, 2–4 July 2019; pp. 1–6.
35. Biggio, L.; Russi, M.; Bigdeli, S.; Kastanis, I.; Giordano, D.; Gagar, D. PHME Data Challenge. *Eur. Conf. Progn. Health Manag. Soc.* **2021**.
36. De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D.L. The Mahalanobis Distance. *Chemom. Intell. Lab. Syst.* **2000**, *50*, 1–18. [CrossRef]
37. Tharwat, A.; Gaber, T.; Ibrahim, A.; Hassanien, A.E. Linear Discriminant Analysis: A Detailed Tutorial. *AI Commun.* **2017**, *30*, 169–190. [CrossRef]
38. Kataria, A.; Singh, M.D. A Review of Data Classification Using K-Nearest Neighbour Algorithm. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 354–360.
39. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. An Introduction to Decision Tree Modeling. *J. Chemom.* **2004**, *18*, 275–285. [CrossRef]
40. Suthaharan, S. Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*; Suthaharan, S., Ed.; Integrated Series in Information Systems; Springer: Boston, MA, USA, 2016; pp. 207–235. ISBN 978-1-4899-7641-3.
41. Wickramasinghe, I.; Kalutarage, H. Naive Bayes: Applications, Variations and Vulnerabilities: A Review of Literature with Code Snippets for Implementation. *Soft Comput.* **2021**, *25*, 2277–2293. [CrossRef]
42. Xu, Q.-S.; Liang, Y.-Z. Monte Carlo Cross Validation. *Chemom. Intell. Lab. Syst.* **2001**, *56*, 1–11. [CrossRef]
43. Kotu, V.; Deshpande, B. *Predictive Analytics and Data Mining: Concepts and Practice with Rapidminer*; Morgan Kaufmann: Waltham, MA, USA, 2014.
44. Daga, A.P.; Fasana, A.; Garibaldi, L.; Marchesiello, S. Big Data Management: A Vibration Monitoring Point of View. In Proceedings of the 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Roma, Italy, 3–5 June 2020; pp. 548–553.