A Self-supervised Classification Algorithm for Sensor Fault Identification for Robust Structural Health Monitoring

Andreea-Maria Oncescu[1] and Alice Cicirello[2(✉)]

[1] Department of Engineering Science, University of Oxford, Parks Road,
Oxford OX1 3PJ, UK
andreea-maria.oncescu@sjc.ox.ac.uk

[2] Department of Engineering Structures, Delft University of Technology,
2628 CN Delft, The Netherlands
a.cicirello@tudelft.nl

Abstract. A self-supervised classification algorithm is proposed for detecting and isolating sensor faults of health monitoring devices. This is achieved by automatically extracting information from failure investigations. This approach uses (i) failure reports for extracting comprehensive failure labels; (ii) recorded data of a faulty monitoring device and the information of the failure type for selecting fault-sensitive features. The features-label pairs are then used to train a classification algorithm, so that when a new set of measurements becomes available, the algorithm is capable of identifying with a high accuracy one of the possible failure types included in the training data set. The proposed approach is successfully applied to the failure investigations conducted on a low-cost wearable device, displaying similar challenges encountered in SHM.

Keywords: Sensor failures · Monitoring device failure · Self-supervised machine learning · Natural language processing · SHM

1 Introduction

Advanced monitoring strategies are developed for tracking the health status of engineering structures (Structural Health Monitoring, SHM) [1,2] and of people [3,4] to make inferences on the health condition and support decisions, such as preventive actions to restore normal conditions. Nonetheless, while the recorded data is enabling the investigation of features, correlations and associational evidence, it is not sufficient for the identification of causal relationships or the distinction of confounding sources. Domain knowledge and the availability of a causal model, enable one to move from accurate-but-wrong predictions purely based on data (obtained from a wrong data-based model and/or corrupted measurements) to explainable and interpretable inferences needed to support decision making. To make inferences on the health condition of a system, measurements must be informative, reliable and accurate. However, during operating conditions, a monitoring device might be prone to unnoticeable failures

P. Rizzo and A. Milazzo (Eds.): EWSHM 2022, LNCE 253, pp. 564–574, 2023.
https://doi.org/10.1007/978-3-031-07254-3_57

caused by poorly manufactured sensors and/or electronics, problems with cable harnesses, ageing effects, improper handling, electromagnetic interference, and environmental factors [5]. This is one of the key bottlenecks undermining the reliable deployment of monitoring technologies. In SHM, a faulty monitoring device could lead to an accurate-but-wrong assessment of the remaining useful life of a structure [5]. In health monitoring devices, it can cause fatal conditions to be missed, over-treated and it might produce health anxiety or fatigue [3,6].

Currently failures of the monitoring device are investigated by (i) periodic inspection; (ii) implementing an additional monitoring system. These investigations are costly and do not ensure the detection of a faulty monitoring system [5]. As a result, several approaches have been developed for automatic sensor fault detection (has a fault occurred?), isolation (what is the type, location and extent of the fault?) and reconstruction and/or mitigation (can the data be corrected to reduce the fault effects?). Broadly speaking, they can be grouped into [5,13]: model-based, knowledge-based and data-driven approaches. These approaches have been used in different application domains: chemical process monitoring [7], aircraft control applications [8,9], wearable health monitoring devices [10] and SHM applications [1,2,5,11,12,23].

The identification of a faulty sensor in SHM is particularly challenging since it would require the identification and distinction of the monitoring failures from structural failures and/or operating and environmental conditions. This is one of the key bottlenecks that cannot be easily circumvented by using only data-driven approaches [14–19]. One way around this is to employ a supervised machine learning approach based on discriminative features in the measurements and directly pair them with failure labels of the structure and/or of the monitoring device. This is extremely challenging, since it would require an engineer to manually and accurately label measurements in real-time [1,2,20] and to identify discriminative features.

This paper proposes a self-supervised classification algorithm for sensor fault detection and isolation that leverages additional information provided in failure investigations of the monitoring devices. In particular, it is based on exploiting the domain knowledge on monitoring device failures by automatically extracting comprehensive failure labels from failure reports (exploiting the techniques developed in [26]). The data recorded with faulty and healthy devices is then used to select discriminative features required to build a training data set of features-label pairs. This self-supervised classification algorithm is trained so that when a new set of measurements becomes available, one of the possible health conditions included in the training data set can be isolated. The feasibility of the proposed approach is shown through its application to the failure investigations carried out on a low-cost wearable device based on an Arduino programmable board with 4 sensors. This application displays similar challenges encountered in SHM: (i) the sensors record various quantities at different rates; (ii) operational and environmental conditions affect the measurements; (iii) a sensor might display similar failure types; (iv) a limited data set of recorded failures is available (117); (v) imbalanced number of elements in the training data set. Three classification approaches are compared: Naïve Bayes, Support Vector Machines and Artificial Neural Networks. Finally, the implications of using Natural Language Processing approaches for extracting labels and discriminative features to train each classification approach within the proposed self-supervised classification algorithm for sensor fault identification are discussed.

2 Detection and Isolation of a Faulty Sensor in Health Monitoring Devices

The proposed approach aims at detecting and isolating a faulty sensor in health monitoring devices with a self-supervised machine learning strategy by automatically extracting information obtained during failure investigations.

Failure investigations of a device, system or structure are carried out by an expert to identify the failure root-cause and propose remedial actions [21,22]. A failure investigation include the analysis of the measurements collected in operating conditions before and after the failure occurred. Moreover, it uses additional laboratory experiments to identify the root-cause of the failure. Once the failure has been identified a failure report is written. A failure report has a standard outline [21,22] and it consists of sections written as free text and images. The failure effects observed during operating conditions are described, together with a brief description of the patterns observed in the measurements. Other sections focus on describing the steps taken to identify the root-cause of failure and to reproduce it in laboratory conditions and the remedial actions to be implemented. Often it also includes a section on how to manage similar failures in the future. Currently, the information collected during failure investigations is used for quality assessment, to support decisions about design changes and schedule maintenance [21,22]. To the best of the authors' knowledge, this information has never been used to detect and isolate the failure of a monitoring device. The proposed approach consists of three steps, as shown in Fig. 1.

Step 1: Data Collection. It is proposed to use the information obtained during failure investigations together with the data obtained in operating conditions.

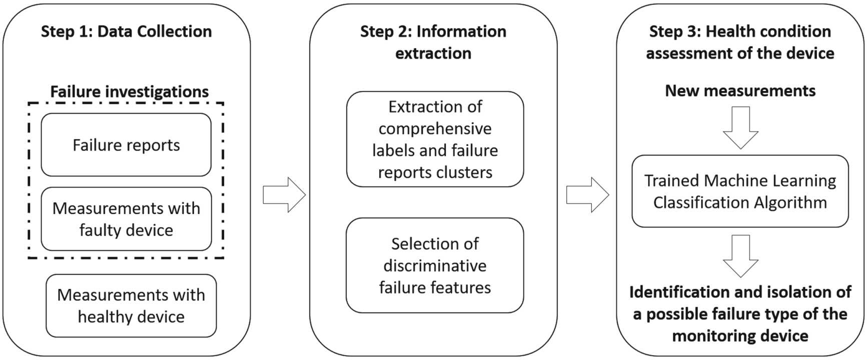
Step 1: Data Collection
Failure investigations
Failure reports
Measurements with faulty device
Measurements with healthy device
Step 2: Information extraction
Extraction of comprehensive labels and failure reports clusters
Selection of discriminative failure features
Step 3: Health condition assessment of the device
New measurements
Trained Machine Learning Classification Algorithm
Identification and isolation of a possible failure type of the monitoring device

Fig. 1. Schematic representation of the proposed approach

In particular, failure reports, measurements obtained with the faulty device and measurements collected with the healthy device are used together to automatically evaluate the relationship between patterns in the data (features) and the failure type (label). In the remaining of the paper, it is going to be assumed that the data from failure investigations is available.

Step 2: Automatic extraction of features-label pairs describing the monitoring device failure types: (i) Labels extraction and failure report clusters identification. The strategy based on Natural Language Processing (NLP) proposed in [26] is employed to extract comprehensive labels from the failure reports; (ii) Feature extraction from measurements. It is proposed to select discriminative features based on the information obtained during the failure investigations.

Step 3: Classification of the monitoring device health condition for a new set of measurements. A classification algorithm is trained and used to detect and isolate one of the possible sensor fault classes included in the training data set. Three standard supervised classification algorithms are investigated [27]: Naïve Bayes, Support Vector Machines and Artificial Neural Networks.

2.1 Review of Labels Extraction and Failure Report Clusters Identification

Manually extracting comprehensive labels from the failure reports can be time-consuming. A strategy consisting of four steps for efficiently identifying the failure types with little input from the user was proposed in [26]:

1. Report to Text: The text is extracted from the failure reports (e.g. by using the *docxpy* python package).
2. Preprocessing: Relevant words only are retained by performing: tokenization; removing stop words; part of speech tagging; and lemmatization.
3. Failure reports as vectors: Term Frequency-Inverse Document Frequency (TF-IDF) [25] is used in combination with Bag of Words (BoW) [25] to refine the list of words. These words and their TF-IDF score are then used to represent each document as a vector by considering those words whose TD-IDF score is above a user-specified threshold.
4. Failure report clustering: Unsupervised (cluster centres randomly allocated) or semi-supervised (initial clusters centre assigned by manually selecting one report from each failure type) K-means clustering is implemented.
5. Failure type label assigned to a cluster: The label is manually extracted from a single report chosen within each cluster. Alternatively, the TF-IDF Centroid algorithm can be implemented.

For small failure data sets, with unbalanced classes and similar failure types, the semi-supervised clustering procedure would lead to more accurate results [26], and would therefore yield more accurate semi-supervised classification results. Nonetheless, it is important investigating if the unsupervised clustering would lead to sufficiently accurate self-supervised classification results.

2.2 Learning Discriminative Features from Failure Investigations

Preprocessing of the raw data is an essential step to reduce the dimensionality of the input vector and to improve the accuracy of the results, as well as to reduce the computational cost required by the algorithm to classify the data. Redundant features can hinder the classification process. We propose to extract features of the measurements linked to each failure type taking also into account the different type of sensors available. These features are critical for differentiating the failure types within the same sensor, and such, they need be robust with respect to the varying operational and environmental conditions.

Broadly speaking the features can be grouped into: (i) statistical time domain features (e.g. average, variance, skew, kurtosis, Root Mean Square); (ii) frequency domain features; (iii) time-frequency domain features (e.g. spectrograms, wavelet transform); (iv) features related to a particular failure type (e.g. number of consecutive zeros or constant values; sudden variations in slow varying sensors); (v) features for distinguishing different failure types of the same sensor. While (i) to (iii) are general, the features in (iv) and (v) are application specific and require domain knowledge. This is because these features capture the known effects of each sensor fault on the measurements (see for example those summarised in Table 2). This domain knowledge is readily available in the failure reports, since expert technicians exploit these features for isolating the failure root cause during failure investigations. Therefore, the discriminative features can be learned by extracting the information from the text in the report and/or from the figures provided. This can be done manually or automatically.

2.3 Classification Algorithms

The three classification algorithms are briefly reviewed. The reader is referred to [27] for more in-depth descriptions.

Naïve Bayes [27] is the most simple technique for constructing a classifier. It is based on assuming statistical independence between the values that certain features observed in the data will take, and it yields the probability of the data belonging to a class conditional on the occurrence of a set of values of these features. In particular, it uses Maximum a Posteriori to assign a class label. This approach requires specifying the number of classes, and the training data set for each class. Moreover, it is usually implemented by specifying a distribution of the likelihood function (usually a Gaussian with mean and variance obtained from the feature values extracted in each class). Naïve Bayes displays good performances even when the amount of training data available is limited. This classification is computationally efficient and therefore, this approach can be readily applied to large test data sets. However, since it assumes independence between features, the results might not be as accurate as when other methods are implemented, such as Support Vector Machines.

Support Vector Machine (SVM) is a very popular non-probabilistic supervised classification approach [27]. It is based on finding a hyperplane that best separates the training data. The parameters of this hyperplane are obtained by solving a convex optimization problem. In particular, the training data set is assumed to be separable in the feature space, so that it is possible to identify a "margin", that is the smallest perpendicular distance between the decision boundary and the closest of the data points. The data points closest to the decision boundary are the so-called support vectors, and they lie on the maximum-margin hyperplanes in the feature space. In its simplest form, the margin is approximated by using a hyperplane in the original features space (linear classification) or by using the so-called "kernel trick" to transform the features space so that a linear separation can still be employed to address a non-linear classification problem [27]. The hyperparameters of SVM are tuned using the cross-validation approach. SVM is usually not efficient when dealing with large amount of data, as compared to other classification algorithms.

The Artificial Neural Network (ANN) is a versatile approach that uses parametric nonlinear functions from a set of input variables to a set of output variables. These functions are defined as linear combinations of nonlinear basis functions and adaptive parameters, the so-called weights [27]. The weights are unknown parameters of the ANN that need to be optimised to improve the accuracy of classification results. The ANN implementation steps are:

1. ANN architecture setup
   Selection of: (i) number of nodes of the output layer (number of classes to be identified); (ii) performance metric (loss function); (iii) activation function for the output layer; (iv) number of nodes of the input layer (corresponding to the input features); (v) number of hidden layers; (vi) number of nodes of the hidden layer; (vii) activation function for the hidden layers.
2. Optimisation setup
   Selection of: (i) optimization algorithm; (ii) learning rate; (iii) batch size; (iv) number of epochs (complete passes through the training data set).
3. Network training
   Steps: (i) Forward propagation: all the training data goes through the network and labels/class predictions are made; (ii) Error in the predictions is assessed by using the loss function; (iii) Backward propagation is used to compute the error for each node of each hidden layer, and to compute the derivatives of the error with respect to the weights; (iv) parameters of the optimization setup, such as the optimization algorithm, learning rate, epochs, batch size and weights initialisation are adjusted to reduce the total loss. Iterations are continued until a sufficient accuracy is achieved.
4. Tuning of Network parameters on validation data set
   The ANN architecture setup parameters, such as the activation function of the hidden layers, the number of nodes of the hidden layers, number of hidden layers are adjusted to reduce the total loss; Iterations are continued until the accuracy does not improve any further.
5. Test on the ANN is performed by using unseen data and assesses the actual (unbiased) performance of the ANN

Compared to the other approaches presented, ANN requires more model parameters to be selected and optimised to increase the accuracy.

Table 1. The seven induced failures and effects on recorded data [26]

Failure type	Effects on measurements	Occurrences
F1 = (GSR, analog, pin)	Jumps to 521 or constant values	24
F2 = (GSR, ground, pin)	Jumps above 1000 or constant values	24
F3 = (GSR, burnt, resistor)	Signal distorted	16
F4 = (accelerometer, ground, pin)	Jumps to higher values	11
F5 = (accelerometer, power, pin)	Jumps to lower values or zeros	11
F6 = (humidity, power, pin)	Jumps to different values or −300%	18
F7 = (temperature, ground, pin)	Jumps to different values or −127 °C	13

3 Case Study: Low-Cost Wearable Device

A low-cost wearable device is chosen for investigating 7 failure types while keeping the costs low. The device consists of a programmable Printed Circuit Board (Adafruit Metro Mini 328), a temperature sensor (digital Dallas Temperature Sensor), a humidity sensor (digital Grove - Temperature & Humidity Sensor Pro), an accelerometer (analog Triple Axis Accelerometer BMA220(Tiny)) and a Galvanic Skin Response (GSR) sensor. The GSR Serial Port Readings are in the range 0–1023, not converted into skin resistance (ohms) in post-processing. The faulty sensors independently investigated, together with the effects on the measurements caused by the failure, are specified in Table 2.

A total of 117 failure instances were manually induced (same as reported in [26]), and an imbalanced number of occurrences was considered for each failure type (Table 1). These failures represent typical failures of small electronic devices [24]. Solder joints and sensor connectors failures were reproduced by disconnecting wires at the interface with the PCB. The burnt resistors failure type was induced by adding a resistor to the analog and power pins of the GSR sensor. Data was recorded during controlled and operating conditions, A failure report was written each time a failure occurred. Data and reports are stored within a Structured Query Language database for easy retrieval of information.

For the label extraction, the same analysis reported in [26] was carried out. Specifically: the TF-IDF score threshold was set to 0.0019. The K-means implementation from the sklearn package [28] was used where the cluster number was set to 7. For the unsupervised K-means clustering with 100 starting points [26], a maximum accuracy of 83.7% was observed, and a lowest of 70.1%. However, it was noted that a failure type (accelerometer, power, pin) could be entirely miss-clustered. A higher accuracy was obtained when considering semi-supervised K-means clustering (assigning one report per failure type), and although a small number of reports was miss-clustered the overall performance of label extraction was able to reach a certain target accuracy while at the same time reducing the setting up time [26]. Based on the information contained in the failure reports, the discriminative features reported Table 2 were considered. A total of 73 fea-

Table 2. Discriminative features selected based on failure investigations

Sensor	Selected features	To detect
All sensors	Average, variance, skew, kurtosis	Time domain variations
Accelerometer	Root Mean Square	Baseline changes
Accelerometer	Consecutive recorded values of zeros	Power pin failure
Accelerometer	First two frequencies peak magnitude	Failure differentiation
Accelerometer	First two frequencies peak position	Failure differentiation
Temperature	$\Delta_{\mathrm{T}} = \max\lvert \mathrm{T}(t_i) - \mathrm{T}(t_{i+1})\rvert > 2\,^{\circ}\mathrm{C}$	Sudden jumps
Temperature	Most Frequent T (T_{MF}) when $\Delta_{\mathrm{T}} > 2\,^{\circ}\mathrm{C}$	Ground pin failure
Temperature	Number of occurrences of T_{MF}	Intermittent failure
Humidity	$\Delta_{\mathrm{H}} = \max\lvert \mathrm{H}(t_i) - \mathrm{H}(t_{i+1})\rvert > 2\%\,\mathrm{RH}$	Sudden jumps
Humidity	H_{MF} when $\Delta_{\mathrm{H}} > 2\%\,\mathrm{RH}$	Power pin failure
Humidity	Number of occurrences of H_{MF}	Intermittent failure
GSR	Maximum value of autocorrelation	Analog/Ground failure
GSR	Minimum value of autocorrelation	Analog/Ground failure
GSR	$\max\Delta_{\mathrm{GSR}}(t_i) = (\max\lvert \mathrm{GSR}(t_i) - \mathrm{GSR}(t_{i+1})\rvert)$	Analog/Ground failure
GSR	$\max\mathrm{GSR}(t_*)$ with $(t_*) = \mathrm{argmax}\Delta_{\mathrm{GSR}}(t_i)$	A/G/R failure differ
GSR	$\min\mathrm{GSR}(t_*)$ with $(t_*) = \mathrm{argmin}\Delta_{\mathrm{GSR}}(t_i)$	A/G/R failure differ
GSR	$\mathrm{GSR}_{\Delta t}$: constant GSR for $\max\Delta t$	A/R failure differ
GSR	length of $\max\Delta t$	Detect GSR failures

tures were considered (considering each direction x, y and z measured by the accelerometer separately), and the min-max normalisation was implemented.

For the classification, the data set includes the 117 failure instances and a set of 15 measurements obtained with a working device (with 13 used for training). Therefore, a total of 8 classes are considered: 7 for each type of failure and one for the working condition. The same training and testing data-split (80–20) is used in each approach, so that the sums of the predicted failure numbers in each row of each the confusion matrices are the same. A total of 100 runs for different splits of the training-validation data sets are evaluated for tuning the hyperparameters of SVM and ANN. The hyperparameter used were: For SVM: kernel = *sigmoid*, $C = 1000$, $\gamma =$ auto and the decision function is one versus one. For ANN: 2 hidden layers, 189 hidden nodes, ReLU activation function, regularizer: 0.00199, learning rate: 0.01703, number of epochs is 159. Early stopping is employed for the ANN, such that the best performing model on the validation data is saved. The results obtained in terms of average accuracy (evaluated across the 100 runs of the algorithms) are summarised in Table 3. The supervised classification refers to the case where all the correct labels are manually assigned, and it is used to assess the robustness of the discriminative features. The semi-supervised and self-supervised classification refers to the use of labels extracted with the semi-supervised and unsupervised K-means, respectively.

Table 3. Average accuracy obtained with three classification algorithms

Classification	Naïve Bayes	SVM	ANN
Supervised	0.926	0.963	0.967
Semi-supervised	0.852	0.898	0.893
Self-supervised	0.704	0.870	0.874

As expected, all the classification algorithms perform better when the labels are manually extracted and assigned by the user. It can also be observed that overall, the SVM and ANN perform better in terms of mean accuracy than Naïve Bayes in all types of learning even if both SVM and ANN can sometimes achieve lower accuracy scores across the different training-validation data split. For the self-supervised case, the ANN yields the best performance while in the semi-supervised case SVM performs best. Overall, Naïve Bayes has provided a good compromise in terms of model set-up, computational cost and accuracy.

4 Conclusions

A self-supervised machine learning strategy that enables the detection of a monitoring device failure, and the identification of the failed sensor and the type of sensor failure has been proposed. This approach uses measurements obtained with a healthy monitoring device, measurements obtained with a faulty device and also failure reports obtained during failure investigations. The process of extracting labels from failure reports is sped up by using an unsupervised NLP strategy and the classification is improved by selecting discriminative features based on failure information. Once the features-label pairs are constructed, Naïve Bayes, SVM and ANN were considered and compared.

A low-cost monitoring device was investigated. A limited data set characterised by four different faulty sensors, two of which displayed multiple failure types and an imbalanced number of failure were considered. Seven failure types were manually induced, and measurements with different sensors were recorded during operating and testing conditions. Failure reports for each failure investigated were written, and paired with the recorded data. Moreover, 15 sets of measurements were recorded with the healthy monitoring device.

It was shown that the discriminative features were robust with respect to the operating and laboratory conditions investigated, with average accuracy of the failure types (supervised) classification above 90% for all the classification methods. Regarding the label extraction procedure, it was shown that when dealing with small failure data sets, with unbalanced classes and similar failure types, that the unsupervised clustering procedure can lead to sufficiently accurate (self-supervised) classification results, yielding a mean accuracy for health classification of the monitoring device of above 70% for all the classification approaches. Semi-supervised classification yielded an average accuracy of above 85%, with ANN and SVM yielding similar performance.

References

1. Farrar, C.R., Worden, K.: Structural Health Monitoring: A Machine Learning Perspective. Wiley, Hoboken (2013)
2. Barthorpe, R.J., Worden, K.: Emerging trends in optimal structural health monitoring system design: from sensor placement to system evaluation. J. Sens. Actuator Netw. **9**(3), 1–31 (2003)
3. Patel, S., Park, H., Bonato, P., Chan, L., Rodgers, M.: A review of wearable sensors and systems with application in rehabilitation. J. Neuroeng. Rehabil. **9**, 1–21 (2012)
4. Dias, D., Paulo Silva Cunha, J.: Wearable health devices - vital sign monitoring, systems and technologies. Sensors **8**, 1–28 (2018)
5. Yi, T.H., Huang, H.B., Li, H.N.: Development of sensor validation methodologies for structural health monitoring: a comprehensive review. Measurement **109**, 200–214 (2017)
6. Academy of Medical Royal Colleges: Artificial Intelligence in Healthcare. Academy of Medical Royal Colleges (2019)
7. Dunia, R., Qin, J.S., Thomas, E.F., McAvoy, T.J.: Identification of faulty sensors using principal component analysis. AIChE J. **42**, 2797–2812 (1996)
8. Van Eykeren, L., Chu, Q.P.: Sensor fault detection and isolation for aircraft control systems by kinematic relations. Control Eng. Pract. **31**, 200–210 (2014)
9. de Silva, B.M., Callaham, J., et al.: Physics-informed machine learning for sensor fault detection with flight test data. arXiv (2020)
10. Zhang, H., Liu, J., Kato, N.: Threshold tuning-based wearable sensor fault detection for reliable medical monitoring using Bayesian network model. IEEE Syst. J. **12**, 1886–1896 (2018)
11. Friswell, M.I., Inman, D.J.: Sensor validation for smart structures. J. Intell. Mater. Syst. Struct. **10**, 973–982 (1999)
12. Kerschen, G., De Boe, P., Golinval, J., Worden, K.: Sensor validation using principal component analysis. Smart Mater. Struct. **14**, 36–42 (2004)
13. Li, D., Wang, Y., Wang, J., Wang, C., Duan, Y.: Recent advances in sensor fault diagnosis: a review. Sens. Actuators A: Phys. **309**, 1–13 (2020)
14. Sohn, H.: Effects of environmental and operational variability on structural health monitoring. Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci. **365**, 539–560 (2007)
15. Kullaa, J.: Eliminating environmental or operational influences in structural health monitoring using the missing data analysis. J. Intell. Mater. Syst. Struct. **20**, 1381–1390 (2009)
16. Kullaa, J.: Distinguishing between sensor fault, structural damage, and environmental or operational effects in structural health monitoring. Mech. Syst. Signal Process. **25**, 2976–2989 (2011)
17. Kullaa, J.: Robust damage detection in the time domain using Bayesian virtual sensing with noise reduction and environmental effect elimination capabilities. J. Sound Vib. **473**, 1–13 (2020)
18. Cross, E.J., Worden, K., Chen, Q.: Cointegration: a novel approach for the removal of environmental trends in structural health monitoring data. Proc. R. Soc. A: Math. Phys. Eng. Sci. **467**, 2712–2732 (2011)
19. Avendano Valencia, L.D., Chatzi, E.N., Tcherniak, D.: Gaussian process models for mitigation of operational variability in the structural health monitoring of wind turbines. Mech. Syst. Signal Process. **142**, 1–23 (2020)

20. Bull, L.A., Rogers, T.J., Wickramarachchi, C., Cross, E.J., Worden, K., Dervilis, N.: Probabilistic active learning: an online framework for structural health monitoring. Mech. Syst. Signal Process. **134**, 1–20 (2019)
21. Perry, M.L.: Electronic Failure Analysis Handbook. McGraw-Hill (1999)
22. Otegui, J.S.: Failure Analysis: Fundamentals and Applications in Mechanical Components. Springer, Heidelberg (2014)
23. Martakis, P., et al.: Smart Struct. Syst. Int. J. **29**(1), 251–266 (2022)
24. Jiang, N., et al.: Reliability issues of lead-free solder joints in electronic devices. Sci. Technol. Adv. Mater. **20**(1), 876–901 (2019)
25. Jurafsky, D., Martin, J.H.: Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall Series in Artificial Intelligence (2000)
26. Oncescu, A., Cicirello, A.: Sensor fault label identification for robust structural health monitoring. In: Proceedings of the 4th ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2021), vol. 28 (2021)
27. Bishop, M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
28. Pedregosa, F., Varoquaux, G., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)