



Universidade do Minho
Escola de Engenharia

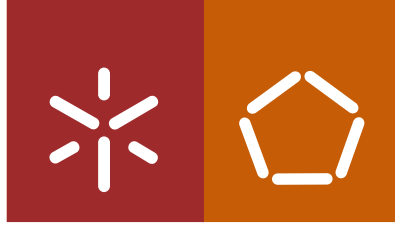
Alberto Barontini

**Bio-inspired algorithms for Structural Health
Monitoring of Civil Engineering Systems**

Alberto Barontini **Bio-inspired algorithms for Structural Health Monitoring of Civil Engineering Systems**

UMinho | 2021

March 2021



Universidade do Minho
Escola de Engenharia

Alberto Barontini

Bio-inspired algorithms for Structural Health Monitoring of Civil Engineering Systems

Doctoral Thesis
Doctoral in Civil Engineering

Work conducted under supervision of
Professor José Luís Ferreira da Silva Ramos
and
Professor Paulo Jorge Rodrigues Amado Mendes
and
Professor Maria Giovanna Masciotta

March 2021

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



<https://creativecommons.org/licenses/by/4.0/>

ACKNOWLEDGEMENTS

The present research was carried out mainly at the Department of Civil Engineering of the University of Minho. Visiting periods at the Department of Mechanical Engineering of the Technical University of Madrid and at the Department of Civil Engineering of the University of Coimbra were also essential for the development of the work.

This work is financed by national funds through FCT - Foundation for Science and Technology, under grant agreement SFRH/BD/115188/2016. Moreover, it is partly financed by FCT/MCTES through national funds (PIDDAC) under the R&D Unit Institute for Sustainability and Innovation in Structural Engineering (ISISE), under reference UIDB/04029/2020.

The research was supervised by Prof. Luis F. Ramos of the University of Minho and co-supervised by Prof. Paulo Amado Mendes of the University of Coimbra and Dr. Maria Giovanna Masciotta of the University of Chieti-Pescara. I would like to express my deepest gratitude to Prof. Ramos for introducing me to Structural Health Monitoring, Prof. Amado Mendes for his support and encouragement and Dr. Masciotta for being not only a mentor but also a friend. I also would like to thank Prof. Ricardo Perera of the Technical University of Madrid for supervising me during my visiting period. Their knowledge and suggestions were incredibly valuable for this work. The experimental work was performed in LEST, Structural Laboratory of the University of Minho. Hence, I would like to thank the technicians, especially Mr. Matos and Eng. Marco Jorge, for their incredible job and Dr. Daniele Brigante for sharing his work in the lab with me.

Finally, my gratitude goes to my mom, dad and grandma, for their continuous encouragement and support to pursue my true passions and goals, and to all the people I met along the way as they helped me become the person I am. In this short space, I would like to recall all of them who experienced my troubles, supporting me in the bad moments and making the good moments memorable. I personally thank Manuel, Michele, Cecilia, Giulia, Serena, Guido and Bianca who saw me grow up, Aldo, whose memory will never fade away, Chandan, Gianpaolo, Carlos and Carmen my first “family” in Guimarães, Luis, Maria Pia, Telma, Elesban, Pilar, amazing colleagues and friends, Antonio, Meera, Rafael and Xinyu, the incredible people I lived with the last part of this experience. Last but certainly not least, Leslie that walking along together with me made this journey more fun and peaceful.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Algoritmos bio-inspirados para Monitorização da Integridade de Estruturas em Engenharia Civil

RESUMO

Hoje em dia, a gestão de um vasto acervo de estruturas complexas e infraestruturas, que se encontram próximo ou para lá do seu fim de vida útil, constitui um importante desafio para os países desenvolvidos. Neste contexto, a manutenção e a prevenção têm vindo a representar custos muito significativos, sendo necessário adotar estratégias de gestão com relação custo-benefício otimizada, mas que ainda se encontram em desenvolvimento. Com a Monitorização da Integridade de Estruturas (em inglês, *Structural Health Monitoring*, SHM) pretende-se garantir a identificação imediata de danos, de forma a permitir uma avaliação automatizada da integridade dos sistemas estruturais. O desenvolvimento desta área de investigação visa a obtenção de métodos adequados e fiáveis para detetar os danos o mais cedo possível, e para que estes sejam encarados de forma imediata, focada e económica. A deteção de danos pode ser formulada como um problema de classificação com uma única classe e pode ser tratada de forma eficaz por meio de ferramentas numéricas bioinspiradas, como o Algoritmo de Seleção Negativa (NSA). A presente tese tem como objetivo desenvolver e testar uma metodologia de deteção de danos baseada numa versão inovadora do NSA com geração determinística. A metodologia é composta por vários recursos numéricos, para ultrapassar as deficiências identificadas na revisão da literatura. A metodologia proposta é validada em análises numéricas e estudos de caso, considerando cenários de danos múltiplos e incrementais, e condições ambientais e operacionais variáveis. Todas as conclusões são baseadas na análise experimental e estatística da aplicação dos algoritmos desenvolvidos, procurando-se desenvolver uma comparação justa com técnicas alternativas. A metodologia proposta revela-se apropriada para a deteção de danos em estado inicial de desenvolvimento. Pode ser adaptada a diferentes tipos de estruturas e propriedades estruturais sensíveis à ocorrência de dano. É robusta em relação a fontes de incerteza como o ruído nos sinais adquiridos, ao erro induzido pela extração das propriedades ou à flutuação devida às condições ambientais variáveis. O seu desempenho é fortemente afetado pela configuração dos parâmetros do algoritmo. Deste modo, são apresentadas diferentes abordagens de configuração, bem como se recomendam valores ou intervalos para parametrização da metodologia. Em conclusão, a estratégia de deteção de danos baseada em NSA, que é validada no contexto da presente tese, é considerada eficaz e os resultados promissores recomendam mais pesquisas e novas aplicações.

Palavras-chave: Algoritmo de Seleção Negativa; Classificação baseada em uma única classe; Deteção de dano; Monitorização da integridade de estruturas; Monitorização de Vibrações.

Bio-inspired algorithms for Structural Health Monitoring of Civil Engineering systems

ABSTRACT

Nowadays, developed countries are challenged by the management of a wide estate inventory of complex existing structures and infrastructures, which are either close or beyond the end of their service life. Maintenance and prevention have become a significant item of expenditure, while cost-effective strategies are required but still under development. Structural Health Monitoring (SHM) aims at the prompt identification of damage in order to allow an automated health condition assessment of structural systems. The development of such a field of investigation shall provide suitable and reliable methods for detecting the damage outbreak at the earliest possible stage, thus for facing it in a quick, focused and economic way. To this end, damage detection can be formulated as a one-class classification problem and effectively addressed through bio-inspired numerical tools, as the Negative Selection Algorithm (NSA). This thesis aims at developing and testing a damage detection methodology based on an innovative version of NSA with deterministic generation. The methodology is composed of several numerical features to tackle the relevant shortcomings that emerged during the literature review and the pilot tests. The individual features and the global methodology are validated on numerical instances and field-testing case studies, considering multiple and increasing damage scenarios and varying environmental and operational conditions. All the conclusions drawn in the present work are based on experimental analyses of the algorithms, performed based on a proper statistical design. Additional attention is paid to provide a fair comparison with alternative existing techniques. The proposed methodology results suitable for early-stage damage detection. It can be adapted to different types of structures and damage-sensitive features. It might be suitable for sensor embedment, by performing the detection on the acquisition of a single sensor. It is independent of the type of monitoring tools or excitation. It is robust against sources of uncertainties as the noise in the signals, the error induced by feature extraction and the fluctuation in the monitored features due to varying environmental conditions. Its performance is, instead, largely affected by the algorithm parameter setting. Therefore, different suitable setting designs are presented together with recommended values or ranges. In conclusion, the damage detection strategy based on NSA, that is validated in the context of the present thesis, is deemed effective and the promising results foster more research and further applications.

Keywords: Negative Selection Algorithm; One-Class Classification; Damage Detection; Structural Health Monitoring; Vibration Monitoring.

TABLE OF CONTENTS

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS	ii
ACKNOWLEDGEMENTS	iii
STATEMENT OF INTEGRITY	iv
RESUMO	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF ABBREVIATIONS.....	x
LIST OF FIGURES	xi
LIST OF TABLES.....	xvi
CHAPTER 1 Introduction	1
1.1. Motivation	2
1.2. Objectives	4
1.3. Thesis outline.....	6
CHAPTER 2 State of the Art	8
2.1. Structural Health Monitoring.....	9
2.1.1. Network design.....	21
2.1.2. Data pre-processing.....	25
2.1.3. Model development	29
2.2. Biomimetics and natural computing.....	36
2.2.1. Biological and artificial immune system.....	39
2.2.2. Nature-inspired SHM	42
2.2.3. Negative selection algorithms (NSA)	47
2.2.4. NSA applications in SHM	58
2.3. Conclusions	59

CHAPTER 3	Deterministically Generated Negative Selection Algorithm (DGNSA)	60
3.1.	Overview of the Damage Detection Strategy	61
3.2.	Algorithm Instance	62
3.2.1.	Deterministic generation	62
3.2.2.	Semi-dynamic DGNSA	66
3.3.	Problem Instance	67
3.3.1.	Monitored Features	67
3.3.2.	Self space approximation	71
3.3.3.	Nonself data required for assessment	74
3.4.	Higher order DGNSA-based damage identification	75
3.5.	Algorithm testing	80
3.5.1.	Goals	81
3.5.2.	Measures of performance	82
3.5.3.	Design of the Experiments (DOE)	89
3.5.4.	Selection of case studies	95
3.6.	Conclusion	98
CHAPTER 4	Application and Validation through Different Case Studies	99
4.1.	Introduction	100
4.2.	Concrete frame building	100
4.2.1.	Description of the case study	100
4.2.2.	Testing campaign	102
4.2.3.	Results	104
4.3.	I-40 bridge over Rio Grande, New Mexico (USA)	127
4.3.1.	Description of the case study	127
4.3.2.	Testing campaign	128
4.3.3.	Results	130

4.4. Conclusions	135
CHAPTER 5 Application to Field-Testing Data	137
5.1. Introduction.....	138
5.2. Church of Monastery of Jerónimos, in Lisbon.....	138
5.2.1. Description of the case study and field work	138
5.2.2. Self set approximation	141
5.2.3. Influence of the DGNSA qualitative and quantitative parameters	152
5.3. Segmental masonry arch.....	161
5.3.1. Description of case study and experiments.....	161
5.3.2. Description of the experimental analysis.....	166
5.3.3. Algorithm validation	167
5.3.4. Algorithm comparison.....	172
5.3.5. Reduced knowledge during training.....	174
5.3.6. Monitoring simulation	177
5.3.7. Conclusion	183
CHAPTER 6 Conclusions and Future Lines of Research	185
6.1. Summary and conclusions	186
6.1.1. Damage detection methodology based on Negative Selection Algorithm	186
6.1.2. Numerical case studies.....	187
6.1.3. Field-test case studies.....	188
6.2. Future lines of research.....	189
LIST OF REFERENCES.....	192
APPENDIX.....	210

LIST OF ABBREVIATIONS

Abbreviations:

AUC	Area Under the Curve
CCD	Box-Wilson Central Composite Design of the experiment
DGNSA	Negative Selection Algorithm with Deterministic Generation
DI	Damage Identification
DOE	Design of the Experiment
EOV	Environmental and Operational Variation
FE	Finite Element
FEM	Finite Element Method
FFD	Box-Wilson Full Factorial Design of the experiment
FN	False Negative result of the classification (actual positive value labelled as negative)
FP	False Positive result of the classification (actual negative value labelled as positive)
MMN	Min-Max Normalisation Algorithm
NDE	Non-Destructive Evaluation
NDT	Non-Destructive Testing
NSA	Negative Selection Algorithm
OSP	Optimal Sensor Placement
QoI	Quantities of Interest
ROC	Reception Operating Characteristic Curve
SHM	Structural Health Monitoring
SMS	Soft Max Scaling Algorithm
TN	True Negative result of the classification (actual negative value labelled as negative)
TNR	True Negative Rate
TP	True Positive result of the classification (actual positive value labelled as positive)
TPR	True Positive Rate
ZS	Z-Score Normalisation Algorithm

LIST OF FIGURES

Figure 1.1: Outline of the thesis.	7
Figure 2.1: Combination of EOVs and damage effect on feature distributions [Modified from [33]] . .	14
Figure 2.2: Example of bivariate data with narrow dimension (modified from [28]).	15
Figure 2.3: Number of sensors deployed, irrespective of their type, in different bridges with maximum length of the bridge span and year of deployment.	16
Figure 2.4: General framework for a Damage Identification strategy.....	19
Figure 2.5: Bivariate data samples projected onto the narrow direction axes (modified from [28]).	28
Figure 2.6: Flowchart of the data-driven and the model-based approaches.....	30
Figure 2.7: Taxonomy of statistical model development divided into the two main approaches and the possible characteristics of the problem. The Rytter's hierarchy with the achievable DI goals and the viable algorithm types for each problem are presented.....	30
Figure 2.8: Summary of the topics analysed in the following sections according to the top-to-bottom order of the NSA components.	50
Figure 2.9: Detector shape in a 2-D feature space, adapted from [193].	53
Figure 2.10: Representation of the generation and censoring stage in a 2-D space, considering a matching rule based on Euclidean distance and a threshold equal to the detector radius.	54
Figure 3.1: Framework of the algorithm, considering the offline parameter tuning and the online dynamic DGNSA.....	61
Figure 3.2: Maximum coverage of parallel and perpendicular lines with a circle.	63
Figure 3.3: Comparison between deterministic (a) and random (b) generations, with the same number of detectors. In the random generation any distribution optimisation lacks.	65
Figure 3.4: Framework for substructured NSA.....	76
Figure 3.5: Examples of the graphs used to report the experiment outcome: (a) occurrence histogram; (b) extraction of the matching chart.....	78
Figure 3.6: Distribution of the anomaly score of a classifier for 4 different classes of damage.....	78
Figure 3.7: Distribution of the anomaly score of two classifiers for 4 different damage scenarios.	80
Figure 3.8: Confusion matrix: possible results of the anomaly detection.	83
Figure 3.9: Example of performance measures: (a) ROC curve; (b) PR curve.....	85
Figure 3.10: Tendency lines of the performance metrics in the analysed case studies.	88

Figure 3.11: Comparison between full factorial design (a) and central composite design (b). The red squares are the levels of the factorial design and the blue circles are the star points. Three and two factor designs are shown.....	91
Figure 3.12: Problem set for pilot study (P), training (R), validation (V) and testing (T) as subset of analyses samples (A) which is also a subset of the entire problem set (D). T is the complementary of the union of V, R and P sets.....	93
Figure 3.13: (a) Estimated marginal means, (b) differences between the means and (c) standard deviations for the three independent variables.....	94
Figure 3.14: Interaction plots combining three factors: (a) factor 1 and 2; (b) factor 1 and 3; (c) factor 2 and 3.....	95
Figure 3.15: Workflow of algorithm validation and testing.....	97
Figure 4.1: Three-story concrete building: on the left, schematic vertical section; on the right, horizontal plan view.....	101
Figure 4.2: Comparison among normalisation algorithms, dataset based on the numerical model of a three-story building, full dataset.....	109
Figure 4.3: Comparison among normalisation algorithms, dataset based on the numerical model of a three-story building, 1 st decimated dataset.....	110
Figure 4.4: Comparison among normalisation algorithms, dataset based on the numerical model of a three-story building, 2 nd decimated dataset.....	111
Figure 4.5: Distribution of the 1 st and 3 rd natural frequencies with their representation in the 2-D space before normalisation.....	112
Figure 4.6: Normalised feature space with D0 and D1 samples for different algorithm instances and log-normal distributions for the two features. In red misclassified damaged samples.....	114
Figure 4.7: log-normal probability density function of the 1 st and 3 rd natural frequencies. In red the false negatives and in grey the probability of the area of the distribution where the false negatives fall. First example: repetition 10 of Min-Max normalisation, repetition 3 of Soft-Max scaling and repetition 12 of Z-score.....	115
Figure 4.8: log-normal probability density function of the 1 st and 3 rd natural frequencies. In red the false negatives and in grey the probability of the area of the distribution where the false negatives fall. Second example: repetition 8 of Soft-Max scaling and Z-score.....	115
Figure 4.9: Training data followed by the validating data, undamaged and damaged in scenario 1 respectively.....	117

Figure 4.10: Training data followed by the test data, undamaged and damaged in scenario 1 respectively.	119
Figure 4.11: Trend of the global classification scores based on averaged sum and weighted sum over the test data.	122
Figure 4.12: Trend of the moving true negative rate over the test data in scenario D1: (a) average over 12 samples; (b) average over 24 samples.	123
Figure 4.13: Trend of the moving true negative rate over the test data in scenario D6: (a) average over 12 samples; (b) average over 24 samples.	124
Figure 4.14: Trend of the moving true positive rate over the test data in scenario D1, compared with: (a) TPR over validating dataset D1; (b) TPR over validating dataset D2; (c) TPR over validating dataset D3; (d) TPR over validating dataset D4; (e) TPR over validating dataset D5; (f) TPR over validating dataset D6.	126
Figure 4.15: Portion of the bridge under investigation, with the location of the damage and the points used to extract the dynamic response.	128
Figure 4.16: Marginal mean plots for the two repetitions of T1.3 (using PSD as signature).	131
Figure 4.17: Matching chart for the second repetition of T1.3, combination 13, using PSD as signature. In the chart, the points which trigger at least one detector are reported in black.	132
Figure 4.18: Histogram of the anomaly score for the algorithm instances tested for higher order classification. The distribution of the anomaly score is based on the validation dataset considering the 2 repetitions for each instance.	133
Figure 5.1: (a) Plant of the Monastery of Jerónimos by Karl Albrecht Haupt, c.1880; (b) Nave of the church. © Wikipedia Creative Commons.	139
Figure 5.2: Plan (a) and cross-section (b) of the church, showing the sensor layout for the continuous monitoring: A (accelerometer), D (datalogger), R (strong motion recorder), TL (tiltmeter), TS (temperature sensor), TH (temperature and humidity sensor), W (anemometer). Schematics of the SHM data workflow (c).	140
Figure 5.3: (a) Trend of temperature and first natural frequency during the monitoring period; (b) Overview of the data samples.	141
Figure 5.4: Trend of the Intersection area (a), the nonself covered area (b), the approximate area (c) and the normalised incremental area (d) for different number of samples.	143
Figure 5.5: Approximate areas by mean of 36 samples compared with the whole dataset in two out of the ten tested repetitions.	144

Figure 5.6: Approximate areas by mean of 54 samples compared with the whole dataset in two out of the ten tested repetitions.	144
Figure 5.7: Approximate areas by mean of 75 samples compared with the whole dataset in two out of the ten tested repetitions.	145
Figure 5.8: Trend of the intersection area (a) and the non-self covered area (b) for different number of divisions and constant data samples.	145
Figure 5.9: Trend of the approximate area (a) and the normalised incremental area (b) for different number of samples adding at each step a single new record to the previous set.	145
Figure 5.10: Approximate area provided by 54 (a) and 67 (b) samples.	146
Figure 5.11: Self dataset and monitored data campaign.	146
Figure 5.12: Trends of the approximate area and incremental area considering the automatic procedure (a-d), manual correction of some approximation (b-e) and daily average (c-f).	148
Figure 5.13: Overview of the data samples. Comparison between the whole dataset and the simulated monitoring campaign with reduced samples.	148
Figure 5.14: Representation of the full self set with its boundary and the artificial non-self set used for performance assessment.	149
Figure 5.15: Performance curves of the different stochastic methods considering different sample sets.	150
Figure 5.16: 54 points approximate areas: (a) deterministic approximation; (b) B-NN-ASR k4; (c) affinity 0.1; (d) affinity 0.2; (e) Gaussian 0.1; (f) Gaussian 0.2.	151
Figure 5.17: Validating self data and validating nonself data in the unitary feature space.	154
Figure 5.18: Comparison between full factorial design (a) and central composite design (b) for two variables. In red the levels of the factorial design and in blue the star points.	157
Figure 5.19: Configuration of the masonry arch.	161
Figure 5.20: Location of the measurement points for the strains (S) and the accelerations (A).	162
Figure 5.21: Location of the three hinges during the test.	163
Figure 5.22: Crack pattern induced by the horizontal settlement.	164
Figure 5.23: Samples under ambient vibration (a) and finger tapping (b).	165
Figure 5.24: Samples under ambient vibration divided in fixed and moving sensors:(a) D0 condition and (b) D1 condition.	165
Figure 5.25: Samples under ambient vibration divided in the different set-ups from (a) D0 condition and (b) D1 condition.	166

Figure 5.26: Feature space for T2 repetition 1 - combinations with low values for all the factors: (a) Training self samples against detectors; (b) Validating self samples against detectors.	171
Figure 5.27: Feature space for T2 repetition 1 - combinations with low value of the detector radius and high values of the self radius and of the censoring distance: (a) Training self samples against detectors; (b) Validating self samples against detectors.	171
Figure 5.28: Feature space for T2 repetition 1 - combinations with high values for all the factors: (a) Training self samples against detectors; (b) Validating self samples against detectors.	171
Figure 5.29: Boxplot of the AUC for the five repetitions of instance T2 and damage scenario D1, analysed through SVM and NSA with different parameters values.....	173
Figure 5.30: Feature space for T2.2 repetition 1, best combination: (a) Training self samples against detectors; (b) Validating self samples against detectors.	176
Figure 5.31: Feature space for T2.3 repetition 1, best combination: (a) Training self samples with variable radius against detectors; (b) Validating self samples against detectors.	177
Figure 5.32: Sequence of the acquisitions for the five repetitions of instance T2, the vertical lines identify the shift to a following damage scenario.....	178
Figure 5.33: Classification success for scenarios D0 and D1. A score equal to 1 means correct labelling, while 0 means incorrect labelling.	179
Figure 5.34: Evolution of the positive counter in the five repetitions of the instance considering different reset values.	180
Figure 5.35: Flow of the acquisitions from the damaged system (scenario D1) considering a threshold of 2: (a) Reset at the 2 nd negative; (b) Reset at the 3 rd negative.	182

LIST OF TABLES

Table 2.1: Compatibility-matrix of functional analogies between biological and SHM systems (cells with dots indicate matching features), modified from [48].	38
Table 2.2: Shortcomings of NSA with specific attention to damage detection.	54
Table 3.1: Pseudo-code of the deterministic NSA censoring stage.	64
Table 3.2: Input-output identification methods (adapted from [245]).	70
Table 3.3: Output only identification methods (adapted from [245]).	70
Table 3.4: Typical performance metrics.	84
Table 3.5: Confusion matrices for different classifiers against the same three case studies.	87
Table 4.1: Analysed damage scenarios, with indication of damage extent and location and of the number of samples for the training and the monitoring phases.	102
Table 4.2: List of analyses carried out.	103
Table 4.3: List of the factor comparisons.	103
Table 4.4: Combination of level for the full factorial design.	104
Table 4.5: Statistics of the ROC AUC with varying maturity age and adaptation rate.	105
Table 4.6: Examples of the variability induced by the three main parameters on the ROC AUC.	106
Table 4.7: Statistics of the ROC AUC considering repetitions with different training sets.	107
Table 4.8: ROC AUC for different scaling algorithms and features.	108
Table 4.9: Statistics for the three different datasets.	111
Table 4.10: Confusion matrix and performance metrics for different scaling algorithms and instances.	112
Table 4.11: Confusion matrix for the best classifier in each feature space.	118
Table 4.12: Performance metrics for the 4 classifiers in the different damage scenarios. For each scenario, the feature space with the best performance is highlighted.	119
Table 4.13: Comparison of the performance metrics per each damage scenario and classifier during the validation and during the test.	125
Table 4.14: Problem instance code for the I-40 bridge case study.	129
Table 4.15: Algorithm instances tested over a larger dataset and confusion matrix.	133
Table 4.16: Classification results for I03.	134
Table 5.1: Synthesis of collected data.	141

Table 5.2: Results of the deterministic approximation of the self space.	143
Table 5.3: Designed data campaign with results. Three colours are used to distinguish the three ranges.	147
Table 5.4: Overview of the statistics. Comparison between the whole and reduced (simulated monitoring) dataset.	149
Table 5.5: Average self set coverage of the different stochastic approximations.	152
Table 5.6: Average nonself set coverage of the different stochastic approximations.	152
Table 5.7: Parameter levels.	154
Table 5.8: Best combination and performance metrics per each test performed. The name of the test refers to the number of training samples and the repetition. In some tests, more combinations had the same best score. Regarding the number of FP and FN it is worth remembering that the total real negatives and real positives are 470. The three best overall performances (for 50, 150 and 300 training samples) are highlighted in bold and grey.	155
Table 5.9: Same of Table 5.8 but with explicit values.	156
Table 5.10: Parameter levels.	158
Table 5.11: Best combination and performance metrics per each test performed. The name of the test refers to the number of training samples and the repetitions. In some tests, more combinations had the same best score. Regarding the number of FP and FN it is worth remembering that the total real negative are 500 and real positive are 520. The six best overall performances (for 50, 150 and 300 training samples on T1 and T2) are highlighted in bold and grey.	158
Table 5.12: Same of Table 5.11 but with explicit values.	159
Table 5.13: Maximisers and maximum of the response surfaces. In bold the best performance per each instance.	160
Table 5.14: Support settlements in the consecutive damage scenarios.	163
Table 5.15: Eigenfrequencies variation over progressive damage scenarios.	164
Table 5.16: Results in the 5 repetitions of the best combinations on the average in instance T1.....	168
Table 5.17: Results in the 5 repetitions of the best combinations on the average in instance T2.....	168
Table 5.18: Results in the 5 repetitions of the best combinations on the average in instance T3.....	169
Table 5.19: Results in the 5 repetitions of the best combinations on the average in instance T2 considering a CCD.	170

Table 5.20: Confusion matrix for the five repetitions of instance T2 and average confusion matrix (ANS and AS indicate the actual nonself and self, respectively; LNS and LS designate labelled nonself and self, respectively).	172
Table 5.21: Results in the five repetitions of the best combinations on the average in instance T2, case 1.	175
Table 5.22: Results in the five repetitions of the best combinations on the average in instance T2, case 2.	176
Table 5.23: Results in the five repetitions of the best combinations on the average in instance T2, case 3.	177
Table 5.24: Number of acquisitions needed to achieve a given positive threshold <i>PT</i> for different reset values (FPA stands for False Positive Alarm).	181
Table 5.25: Probability of firing an alarm within at a given acquisition between 1 and 6 for different combinations of threshold and reset values.	181

CHAPTER 1

Introduction

1.1. Motivation

Engineering systems (facilities, structures and infrastructures) are complex and unique sets of components and materials, operating in different conditions. Due to their influence on human life and activities, it is necessary to guarantee, along their service life, an adequate level of functioning and safety. However, any of these systems is inherently affected by material defects that, although not compromising their ability to meet the requirements they are designed for, in the long term can foster and trigger damage, decay or degradation. The damage can be tolerated as long as the system state is fit-for-purpose. When the level of functioning further decreases, this suboptimal operating condition becomes unacceptable (e.g. in terms of stability, strength or deformation) and it is called failure or system fault [1], [2]. Experience demonstrated that, even when the damage is a localised phenomenon, it can lead to structural global failure [3]. Since the fault is an evident state, the challenge is not to distinguish whether the system reaches or not such a condition, but to warn whenever a damage is detected. Indeed, in many situations, even a sudden failure is the result of a middle-long term process that could have been more readily identified. Traditional methods for structural management mostly rely on regularly scheduled visual inspections, eventually supplemented by non-destructive local tests, like sonic and ultrasonic tests, radiography, eddy current or thermal field [4]. Visual inspections are relatively low cost and simple, however, they are subjected to human error and affected by personal perception and experience. Both research studies and interviews with expert professionals pointed out the large variability of inspection results [5], [6]. These vary, indeed, from inspector to inspector and from inspection to inspection, meaning that even the same inspector under different circumstances can provide a different evaluation of the same item [6]. This variability is likely to be even more significant when dealing with the assessment of the structural response against dynamic loading, as compared to measuring properties in the static range [6]. Moreover, contingent factors such as lack of accessibility, adverse weather conditions, lack of natural light and inappropriate equipment [7] can also affect the process. Accessibility for inspection purposes is a major concern. Many structural elements are hard if not impossible to reach during a routine inspection, being for instance underground, underwater or simply under architectural finishes or covering. It is for instance the case of scour detection in bridge management [8], [9]. This holds not only for normal operations, but also in emergency situations. For example, in post-disaster scenarios, the safety assessment of buildings by visual inspection becomes extremely costly, time consuming and potentially inaccurate due to the non-structural surface coverings that prevent access to the structural elements [7]. Other elements, although exposed cannot be safely reached, as for instance the top chord of high steel trusses bridges under extreme weather conditions. For such elements, the inspection must

be delayed or its quality decreases significantly [6]. Finally, even in optimal conditions, these methodologies allow to identify damages only when circumscribed or superficial, or when information about damage location is already available.

For private and public owners of large asset stocks, the system health assessment should be a major concern, especially in developed countries, where many structures and infrastructures are by now close or beyond their design life, thereby requiring either rehabilitation or downgrading if not able to support today's loading conditions. Due to their considerable number, the immediate replacement of such assets is not feasible, but the consequences of their failure are equally not affordable, due to the risk of injuries and life losses, the relevant economic costs and the repercussion on a higher network level.

Bridges offer a clear example of this, being one of the most relevant structural typologies among today's engineering systems. Their maintenance largely affects the economy of a country, as many reports revealed in the recent past. Administrations in many developed countries (e.g. USA, Australia, UK, etc.) identified a number of deficient bridges, in some cases equal to 40% of the total [10]. In 2002, the Highways Administration (FHWA), in the USA, estimated that rebuilding or rehabilitating the 10000 bridges in need each year would cost 7 billion US\$ per year [10]. In 2003 it was estimated that, in Europe, repairs account for 40% of total construction contract costs [10]. These costs must be compared with the costs of failure, additionally increased when also the secondary consequences are considered as, for instance, in terms of total time travel, increased use of nearby infrastructures and losses of economic benefits associated with the trip no longer available [11]. These secondary societal costs drastically exacerbate the consequences of the failure of transportation network significant assets. However, they make unaffordable even the loss or the closure of very small rural low-volume bridges (average daily traffic of 25 or less vehicles) [12].

Unexpected early failure or loss of serviceability, in the field of Civil Engineering, are making technicians aware of these risks and of the importance of a correct management and increasing their interest in sensing technologies and monitoring strategies to support the inspections [6], [13]. Moreover, the attention of the public opinion towards inspection and monitoring always escalates after catastrophic events, putting owners and managers under investigation and driving both the authorities' efforts to produce better standards and the scientific community's efforts to deepen the understanding of the occurred phenomena and develop new advanced solutions. Nonetheless, large-scale installation and maintenance of monitoring systems are still expensive. Operating in outdoor harsh environment further increases the costs for protection and durability. Unfortunately, installing an insufficient number of sensors to reduce the expenses might turn out to be counter-productive, providing poor information for a

proper assessment. This set of problems result in a general scepticism of the owners towards the monitoring system, considered too expensive, interesting for the research but not truly useful in practice: an attitude that dates back at least to the '80s and it is probably not completely eradicated [6]. For many modern assets, the owners or managers would never endorse to spend more on their monitoring than on their full replacement [6]. However, there exists a class of structures whose replacement is not an option: it is the case of heritage buildings that are prominent historical and cultural symbols with a unique value as assets. For these systems, so many sources of uncertainties can affect a correct structural assessment that sensing technologies might also support the acquisition of a proper level of knowledge to identify and tackle inherent vulnerabilities.

While not claiming to be exhaustive, the examples discussed above prove that structural failure is not affordable and traditional inspection techniques may be inadequate to prevent it. Structural Health Monitoring (SHM) is a field of research concerned with the development of advanced automated methodologies, based on sensor networks and data processing, for the structural integrity assessment. It is possible to speculate that a proper monitoring system would be beneficial for the aforementioned cases and, sometimes, its advantages can be clearly qualitatively identified. Nonetheless, a correct quantification of the value of the structural health monitoring information in real-world applications is still an open issue [14]–[18]. Researchers working in SHM field play an essential role in the definition of innovative automated methodologies to assess the structural health of a monitored system in real-time, without impacting on its operations. Such methodologies should tailor the information granularity and content to the needs of the engineers involved in inspection and maintenance to support their informed decision-making. Moreover, they should be cost-effective and aimed at optimising type, number and location of the sensors to be more appealing for owners and justify their investment. The challenge is clearly to support a new proactive condition-based approach, able to perform early damage detection, avoid catastrophic failures, minimize downtime and reduce human errors in the assessment process.

1.2.Objectives

The main objective of the present study is the development of a reliable early-stage damage detection strategy for Civil Engineering structures, by applying nature-inspired computing techniques, with emphasis on the Negative Selection Algorithm (NSA). This numerical method is selected since it is easy to implement, yet very effective for solving one-class classification, namely the mathematical problem underlying damage detection. The specific goal of the thesis is cast into the long term objective of building up a holistic general framework for SHM, able to address each task, from the network design to the data

processing and the decision making, by optimising costs, complexity, quantity and relevance of the information produced, in order to maximise the value of the monitoring system.

The effective damage detection methodology that is sought must fulfil the following requirements that are specific goals of the present work: (1) the methodology must be able to identify small damage extent, even if the damage is not located in the vicinity of the sensors; (2) it must be robust against typical sources of uncertainties like the noise in the signals or the errors in data processing and feature extraction; (3) it must be able to distinguish and ignore the variations in the monitored features that are induced by operational and environmental conditions, in order to warn only in case of damage; (4) it must be flexible to adapt to different types of structure and different monitored features, both in the case of SHM systems that are installed permanently across the structure for a continuous monitoring and in the case of instruments that are deployed and dismantled after every record; (5) it must provide an accurate detection even in case of limited knowledge, viz. few preliminary data belonging to a single, reference state of the system and no information regarding the damaged behaviour.

In order to achieve such results, initially, a state-of-the-art review of SHM and Natural Computing applications to Damage Identification (DI) is provided. This includes a fully comprehensive framework embracing all the tasks that are involved in DI, together with a critical analysis of the practical shortcomings found in NSA applications.

Based on the identified open issues, an improved version of the NSA with deterministic generation is developed and described. The algorithm is the core of a damage detection strategy which includes other numerical features designed to pursue the aforementioned goals. All these features are discussed along with the general framework of the methodology. Due to the relevance of numerical testing procedures for the validation of the damage detection strategy, a short guideline for algorithm testing is also provided.

Afterwards, the methodology is validated through the application to two numerical case studies: a concrete frame building and a well-known 3-span concrete bridge, i.e. the I-40 bridge. Indeed, artificial case studies allow a full control over all the parameters that might affect the structural response and the performance of the damage detection. As for the first case study, an *ad-hoc* developed instance is used as preliminary test. The damage detection strategy is broken down into its main features. Among them, it is worth mentioning the data scaling, the monitored quantity selection, the improvement of detecting performance in case of scarce and sparse training datasets, the algorithm parameter setting and the design of the experiments. Alternative approaches for each of them are compared in order to identify outperforming solutions and weak points to be tackled. As for the I-40 bridge, being this model calibrated using a real-

world benchmark case study, the quality of the methodology is fully assessed taking advantage of a more realistic instance.

Once the methodology is validated against numerical instances, its effectiveness is tested against two real case studies: a masonry arch built and tested in the laboratory of the University of Minho and a masonry church located within the Monastery of Jerónimos, in Lisbon. In field-testing data, the effect of the sources of uncertainty is more significant, therefore, the robustness of the approach can be better analysed. To this end, a description of the most relevant aspects of the case study, the testing protocol and the operational and environmental conditions during the acquisition is provided. Different solutions for the most sensitive algorithm features, identified during validation, are compared and tested.

A critical assessment of the work is finally performed, allowing to draw interesting conclusions and stress on promising results. Based on that, appealing future lines of research are outlined.

1.3. Thesis outline

In order to pursue the objectives mentioned above, this thesis is organised in six chapters, including this brief introduction, as follows:

- **Chapter 1** is the introduction to the work with the motivation, the focus of the thesis and its outline.
- **Chapter 2** presents the state of the art of Structural Health Monitoring (SHM) for Damage Identification (DI). The main tasks of DI, namely monitoring network design, data pre-processing and model development are presented. Nature-inspired computing methodologies are also introduced as effective approaches for SHM, with specific attention to Negative Selection Algorithms (NSAs).
- **Chapter 3** aims at describing the main core of the present thesis work, namely a complete damage detection strategy based on the development of a new version of the NSA. The features of the methodology are described with particular focus on the main issues identified in the state of the art.
- **Chapter 4** reports the validation of the developed methodology against two numerical case studies: a concrete frame building and the I-40 bridge. The concrete frame building is an *ad-hoc* developed instance, while the bridge model is calibrated though the experimental data from a real-world well-known benchmark case. The high control, typical of numerical simulations, allows to test the performance of the components of the DI strategy based on the proposed NSA. Issues

addressed are the data scaling strategies, the damage-sensitivity of the monitored features and the algorithm parameter setting.

- **Chapter 5** analyses the application of the methodology to two real case studies: a masonry arch built and tested in laboratory environment and a full-scale church exposed to real environmental conditions. Each structure is first presented, then the monitoring system and the in situ acquisition process are described. In the case of the arch, the main aspects of the testing protocol are reported.
- **Chapter 6** discusses the main conclusions of the thesis work and presents a proposal for future research topics.

A schematic representation of this thesis main body outline is displayed in Figure 1.1.

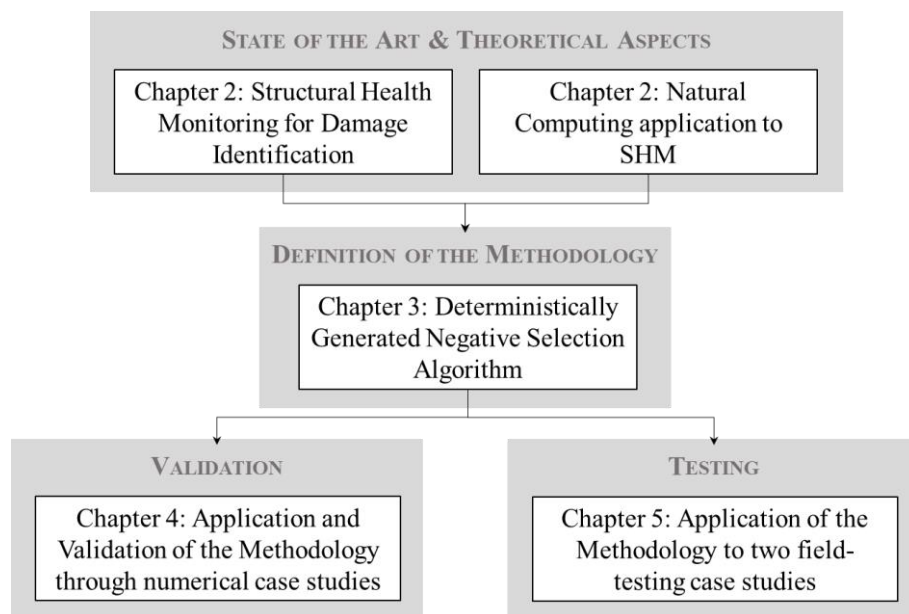


Figure 1.1: Outline of the thesis.

CHAPTER 2

State of the Art

Abstract

This chapter presents a state of the art about Structural Health Monitoring (SHM), focusing on Damage Identification (DI). The main goal is to provide a fully comprehensive framework which collects all the tasks that are involved in DI and stresses all the connections among them. The in-depth discussion of such tasks allows the identification of the main open issues in the field.

2.1. Structural Health Monitoring

Structural Health Monitoring (SHM) is an open multidisciplinary field of research whose main aim is to assess the condition of an investigated structural system. The assessment is carried out based on continuous or periodically sampled acquisitions of some variables, intrinsic or external to the system itself, such as:

- Kinematics: displacements, velocities, accelerations;
- Mechanics: forces, strains;
- Environmental: temperature, humidity, wind pressure and speed;
- Operational: traffic volume, vehicle weight and speed.

Such measurements are used to extract one or more damage-sensitive features and to track their evolution over time. Typical fields of application are Aerospace, Civil and Mechanical Engineering. SHM can be included in a wider Non-Destructive Evaluation (NDE) or Non-Destructive Testing (NDT) field which collects all the non-invasive procedures to assess a system state and measure its characteristics without inducing damage to the system itself nor disassembling any of its parts. Many traditional techniques which are used to inspect Civil Engineering products belong to NDE. Among the most known testing strategies it is worth mentioning Visual, Penetrant, Magnetic Particle, Radiographic, Sonic and Ultrasonic, Eddy Current, Thermal Infrared, etc. The literature on the topic is rather extended, encompassing also several handbooks [19]–[23]. However, SHM outdistances such methodologies in some aspects. First of all, the SHM does not require any previous knowledge of the damage vicinity to carry out a reliable assessment. The aforementioned techniques, instead, to be effective, must be applied in the vicinity of the damage. Due to that, SHM approaches can assess the condition of inaccessible system components and can be applied in real-time and during the system operation. Moreover, the final goal of SHM is to reduce or eliminate the human interaction and judgment component, reaching the so-called intelligent SHM, by defining automatic assessment strategies able to deal with large and complex quantities of data. Finally, the traditional NDE can be economically applied only to few elements within a structural system. In this regard, SHM can work both at the local and at the global level. Global methods assess the system overall state by recording measurements at only few points. The advantages in terms of costs are paid in terms of effectiveness for localisation and quantification. Local SHM techniques, instead, are generally expensive but allow to overcome the aforementioned drawbacks.

A set of 8 general principles holding for any SHM application has been provided by [24], [25]:

- *Axiom I: all materials have inherent flaws or defects.*
- *Axiom II: the damage assessment requires a comparison between two system states.*

- *Axiom III: identifying the existence and location of damage can be done in an unsupervised learning mode, but identifying the type of damage present, and the damage severity can generally only be done in a supervised learning mode.*
- *Axiom IVa: sensors cannot measure damage. Feature extraction through signal processing and statistical classification is necessary to convert sensor data into damage information.*
- *Axiom IVb: without intelligent feature extraction, the more sensitive a measurement is to damage, the more sensitive it is to the environmental and operational conditions variation (EOV).*
- *Axiom V: the length and time scales associated with damage initiation and evolution dictate the required properties of the SHM sensing system.*
- *Axiom VI: there is a trade-off between the sensitivity to damage of an algorithm and its noise rejection capability.*
- *Axiom VII: the size of damage that can be detected from changes in system dynamics is inversely proportional to the frequency range of excitation.*
- *Axiom VIII: damage increases the complexity of a structure.*

Although they might not form a complete set of axioms, it is a first attempt to formulate a general SHM theory. The initial state of the system, also called normal, healthy, undamaged or self, is considered the reference one. Any system, even the most perfect, presents flaws (Axiom I). They can be voids, vacancies, inclusions, impurities or dislocations at the nano/microstructural level, sometimes due to manufacturing. Although the defect is not a problem itself, the local material resistance is reduced and under specific loading conditions a localised crack can appear and evolve at the component level and then at the system level [24], [25]. Irrespective of the inherent material defects, the system can operate with an acceptable quality, where quality is defined as the system ability to meet its requirements. When the defect leads to a state of damage, decay or degradation, the system suffers a reduction of quality and moves from the ideal to a suboptimal operating condition. This is called abnormal, unhealthy, damaged or nonself state. However, the quality can still be satisfactory, this is called normal operating condition, as long as there is an assurance that the system is in a state fit-for-purpose [1]. Finally, the damage scenario can reach an extent where the reduction in the system quality is unacceptable, namely the system does not operate satisfactorily anymore (e.g. in terms of stability, strength or deformation), called failure or state of system fault [2]. Since the failure is an evident state, the challenge is not to distinguish whether the system reaches or not such a condition, but to warn whenever a damage which might lead to failure is detected. A potential damaged condition is called anomalous, until the actual state is confirmed.

Assessing the system condition is a challenging task also because a univocal definition of damage does not exist. Farrar et al. [26] provided a broad definition of damage as any detrimental change intentionally or unintentionally occurred to a structural system, including material and geometric properties, boundary conditions and system connectivity, which deteriorates its current or future performance. The damage always begins at the material level, progressing then at the component and system level (length scale progress). Moreover, based on the characteristics of the damage and of the loading conditions, its effect on the system performance, namely the detrimental change, can be both sudden (e.g. due to extreme discrete events etc.) or accumulate incrementally over a long time (e.g. fatigue, corrosion, ageing, etc.) (time scale progress). The extreme discrete events can be both scheduled or unscheduled and both man-made or natural [24], [25]. Irrespective of the definition, a direct measure of the damage is impossible. Thus, SHM theory assumes that there exists a set of intrinsic features of the investigated system whose magnitude changes when the system condition changes. Such features are sensitive to damage onset and/or progression and represent the state of the system (Axiom IVa). The variables measured directly by the monitoring system are called observables and any observation is a raw dataset, whereas the damage sensitive variables, called Quantities of Interest (QoI), can be either the observables or correlated to and extracted (indirectly measured) from them.

The basic devices used in SHM to measure the observables are the sensors. Many of them exist with different purposes. However, for a general theory of the SHM, it is defined as logical sensor, or simply sensor, any source of information [27]. The characteristics of the sensing system to be designed strongly depend on the length and time scale progress of the expected damages (Axiom V), namely the damages which one expects to identify through the monitoring system. The characteristics of the analysed dynamic signal, in fact, limit the extent of damage which can be detected (Axiom VII), considering also the noise that increases the signal variability reducing the detectability (Axiom VI). More details regarding these aspects are provided in section 2.1.1.

Therefore, assuming that the system condition is described by a vector $\mathbf{y}$ and given a feature vector $\mathbf{x} \in \mathbf{X}$, the SHM theory is based on the existence of a function f such that:

$$\mathbf{x} = f(\mathbf{y}) \quad (\text{Eq. 2.1})$$

In agreement with [28], $\mathbf{x}$ is called point and it is a multidimensional vector. The domain of the points is called input space, namely the space of the domain of each single feature in the vector. For instance, if the point is a m dimensional vector and each of the m features can vary over the entire real number set, the input space $\mathbf{X}$ is $\mathbb{R}^m$. The identification is carried out on a subset of the whole input space $\mathbf{F} \subseteq \mathbf{X}$ called feature space [28]. Each point $\mathbf{x}_i = \{x_{ij}\}_{j=1}^m, i = 1, \dots, L, j = 1, \dots, m$, in the m dimensional

feature space, is extracted from a single measurement. Given a set of L measurements, $t_i, i = 1, \dots, L$ is the initial time of each of them. To perform the identification, the duration $\Delta t_i, i = 1, \dots, L$ of the measurements must be short enough compared to the time scale of the feature physiological or pathological change.

Damage Identification (DI) is a subdiscipline of SHM. However, the overlapping between their goals is so strong that the SHM ultimate purpose has been described as the process of implementing DI strategies [2]. The DI process can be summarized according to Rytter's hierarchy in five main goals of increasing complexity [29], [30]:

- (1) Detection of existence;
- (2) Localisation;
- (3) Classification of the type;
- (4) Quantification of the extent;
- (5) Prognosis.

The hierarchical structure requires that all the lower levels are available before achieving a new level of information. The cumulative information produced by these consecutive steps allows to plan proper interventions to restore the best condition or at least to ensure safety [1].

Considering the above taxonomy, the damage vector can be described, in the most general case, as:

$$\mathbf{y}_i = \{c, l, e\}_i \quad (\text{Eq. 2.2})$$

where $\mathbf{y}$ is the system state vector which collects the information regarding the class of damage c , its position l within the system and its extent e . Different strategies can be defined for the formulation of such information, considering both scalar and vector variables. For the simple detection (level 1) $c \in \{\text{Nonself}, \text{Self}\}$, whereas for higher order classifications (level 3) a set of expected classes must be defined, one for the undamaged state and the others for the damage scenarios. l and e can be collected into a single vector, whose elements' number is equal to the number of subdivisions of the system, following an order depending on the location within the structure, and whose elements' value is equal to the damage index for the corresponding system portion. The domain can be discrete, countably infinite, uncountably infinite or a combination of them. However, it is convenient to define the state variable on a discrete domain $\mathbf{Y}$ of size $|\mathbf{Y}| = N_Y$, for instance by mapping each combination of N_l possible damage locations and N_e possible damage extents onto the natural numbers (e.g. $y_0 = \{\emptyset\}$, no damage, $y_1 = \{l_1, e_1\}, \dots, y_{N_e} = \{l_1, e_{N_e}\}, \dots, y_{N_l \cdot N_e} = \{l_{N_l}, e_{N_e}\}, y_{N_l \cdot N_e + 1} = \{(l_1, e_1), (l_2, e_1)\}$ and so on considering more damaged elements at the same time). The resulting states are $\mathbf{y}_i, i = 1, 2, \dots, N_Y$ mutually exclusive and exhaustive scenarios [31].

Whether performed according to a supervised or unsupervised learning mode, the damage assessment is always based on a comparison between the actual state and the reference state (Axiom II). Therefore, the discrimination between a normal state and an abnormal or damaged state is the fundamental problem of DI, and it is based on one or more metrics, namely quantities that define the similarity or the difference between two feature vectors extracted from two different datasets, in different time instants [32]. Recalling (Eq. 2.1), given a reference state $\mathbf{y}_0$ and an actual state $\mathbf{y}'$, the assessment consists in estimating and comparing $\mathbf{x} = f(\mathbf{y}_0)$ and $\mathbf{x}' = f(\mathbf{y}')$. The damage detection is basically a measure of the discordance between indices representing the healthy system and new indices representing the actual state. An estimate of the discordance is then compared to a threshold set to distinguish satisfactory behaviours and faults, which require a strict quantified definition of failure. The definition of the threshold itself needs the reference state knowledge. One of the main problems in SHM, related to the need for a baseline, consists in the sensitivity of the reference state not only to damage but also to other parameters. Recalling ((Eq. 2.1) and introducing an independent variable, $\mathbf{v}$, which accounts for the EOVs that affect the feature, the function governing the feature value is:

$$\mathbf{x} = f(\mathbf{y}, \mathbf{v}) \quad (\text{Eq. 2.3})$$

This problem, defined by (Axiom IVb), can be easily explained adopting an example, modified from [33]. Let $\mathbf{x} = \{x_1, x_2\}$ be a point composed of two damage-sensitive features and let $\mathbf{y} = \{\text{undamaged}, \text{damaged}\}$ be the possible states. Assume that the features are also affected by the EOVs, through the single factor $\mathbf{v} = \{v\}$. The dependency on v induces a shift in the mean and variance of the feature distribution, namely the ellipse representing the distribution in Figure 2.1 moves within the feature space, changing the location of the centroid, the shape and the size (e.g. from 1 to 4 in Figure 2.1). Whenever a damage emerges there are two extreme cases. Given a change induced by damage analogous to a change produced by EOVs, the shift in the feature distribution is completely masked. Given, instead, a change induced by damage, in some way orthogonal to the change induced by EOVs, the distribution is moved to a location where it is easily distinguished as anomalous.

Thus, it is required the so-called data normalisation to support any conclusion regarding a significant damage-induced discordance between the actual and the reference state. The aim of data normalisation is to convert the feature of (Eq. 2.3), into the following:

$$\mathbf{x}_n = f(\mathbf{y}) \quad (\text{Eq. 2.4})$$

namely it is a projection of the points from the original feature space of $\mathbf{x}$ onto a new feature space, achieving a new normalised feature vector $\mathbf{x}_n$ with strong $\mathbf{y}$ -dependence and weak or null $\mathbf{v}$ -dependence.

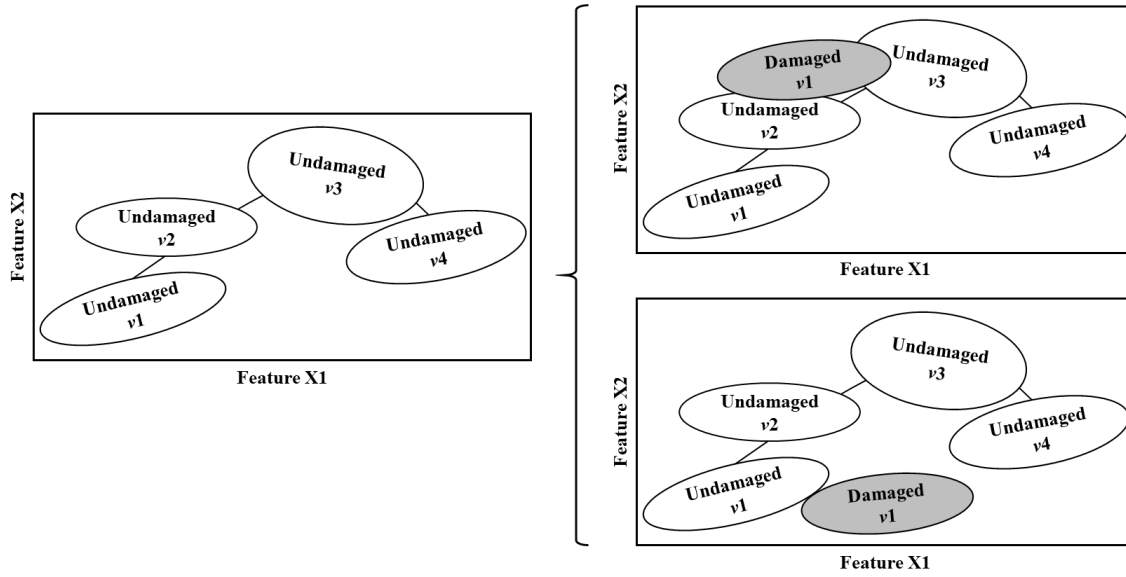


Figure 2.1: Combination of EOVs and damage effect on feature distributions [Modified from [33]].

The DI problem is finally formulated as an inverse problem, whose aim is to calculate, from a set of monitored features, the causal factor $\mathbf{y}$, namely, the function f^{-1} :

$$\mathbf{y} = f^{-1}(\mathbf{x}_n) \quad (\text{Eq. 2.5})$$

In the following, the point $\mathbf{x}_n$ is called $\mathbf{x}$, for the sake of simplicity. A solution to the problem is found by defining a function or a process g which models the actual inverse function, f^{-1} , based on a learning process which uses data previously acquired from the system. This process, called training or calibration depending on the numerical approach followed (see section 2.1.3), consists in finding the best setting for the parameters of a class of models, where the model is the correlation map between the features and the state. Due to the sources of uncertainties, the map f produces more than a single point in the feature space for each system state. All the points belonging to the same i -th system condition $\mathbf{y}_i$ occupy a region S_i . Whenever the statistical distribution of $\mathbf{y}_i$ is multimodal, namely with several peaks or in general well-defined clusters, the region belonging to the same state can be decomposed in different sub-regions (as in Figure 2.1) and the identification can be carried out separately. Otherwise, whenever the sources of uncertainties induce continuous changes and the region is univocal, according to Axiom IVb the detection is more effective along the narrow dimensions of the distribution (see Figure 2.2), assuming that the damage has important projections in these directions [28]. It is assumed that a sharp boundary, Γ_{S_i} , can be determined for each region as the ellipse in Figure 2.2, irrespective of a fuzziness in the boundary area due to the overlapping between the distributions. This is also called decision boundary and it is used as a threshold to separate the samples and to identify which one of the different eventually overlapping classes they belong to. Recalling the Axiom II, the identification is based on a measure of the

distance between the actual image and the pre-damaged image, namely the position that the damaged point would have assumed in the feature space if the system was undamaged [28].

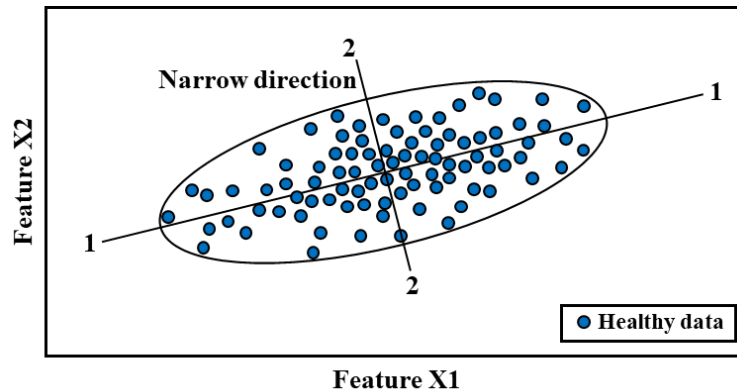


Figure 2.2: Example of bivariate data with narrow dimension (modified from [28]).

A good identification method provides an evidential response, namely it provides information about the response and the confidence in the response [34]. The identification of the state, thus, is provided with a pre-defined level of confidence [28]. This can be done by analysing the points as events of a multivariate random variable, thus estimating the probability density of each state distribution and casting the identification in a probabilistic framework.

The SHM axioms and theory must support the definition of the DI strategy to be applied to a specific case study. The following requirements of a SHM system are adapted and extended from a list reported in [35]:

- Low cost;
- Suitable for continuous, on-line assessment;
- Sensitive to low levels of damage;
- Sensitive to different damage types;
- Insensitive to operational/environmental loading;
- Insensitive to measurement noise;
- Insensitive to operational/environmental condition changes.

According to [35], existing SHM methodologies are able to fulfil only one or a few of the afore-mentioned requirements, thus, more research is needed to develop strategies able to meet them all. However, satisfying all these requirements implies the theoretical efficacy of the system, but not its effectiveness for a real-world application. To be suitable and effective, a system must obtain consistent performances irrespective of the size and the quality of the analysed datasets, namely it must be:

- Robust against errors during the execution of the strategy and erroneous inputs;
- Scalable to avoid performance deterioration (in terms of accuracy, time requirement, etc.) due to growing amount of work.

Nowadays, in fact, large acquisition rate and dense sensor networks can determine data inundation and overload. As demonstrated by [36], well-known algorithms used for system identification and DI (ERA-OKID-00, Single Variate Regression, Autoregressive and Auto Regressive with Exogenous input models) proved to be not scalable, in terms of CPU run time, with linear increase of the number of sensors and/or of the number of samples. The algorithms were particularly sensitive to the number of sensors. Even common pre-processing procedures, like data load, detrending, down sampling or Fast Fourier transform (FFT), undergo severe cost growth with the problem dimension. Data load, FFT and damage identification algorithms even showed a significant nonlinear behaviour [36].

Figure 2.3, modified and extended from [37], shows the increment in the number of sensors deployed for the monitoring of large systems (e.g. high buildings and long bridges). For a lack of information, it is not possible to report the exact number of acquisition channels for each of them. Regarding the Guangzhou TV Tower [38] the number of sensors sums up the in-construction and the in-service systems, counting only once the sensors used for both. The specific applications are further described in the following papers [38]–[43].

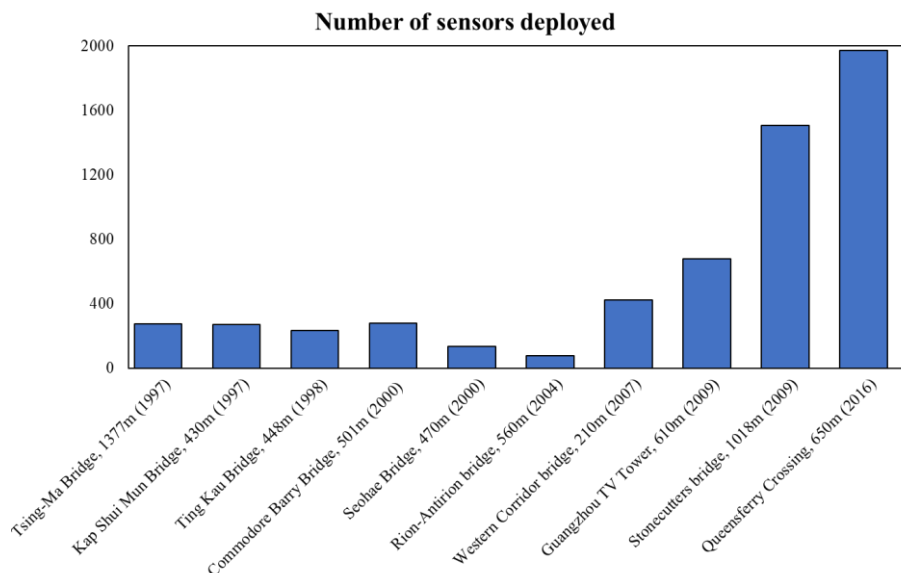


Figure 2.3: Number of sensors deployed, irrespective of their type, in different bridges with maximum length of the bridge span and year of deployment.

In the last three decades, indeed, the technology development has not only improved the SHM prospects through innovative high-performance sensors and hardware, but it has also drastically increased the

challenges. Large sensor networks generate datasets which are growing in size and complexity. The processing units need to manage information which are collected from different type of sensors, with different sampling rates and in varying conditions. The quality of the data is often doubtful and likely to be incomplete and with missing parts [37]. Practitioners and scholars who take part in the development of this field of research should have clear in mind that the final goal is to increase the value of the SHM. This is partially achieved by the improvement in the single components. Nowadays, the single components and strategies (e.g. sensing unit architecture design, sensor number and placement optimisation, large data real-time management) exist and are extremely advanced. Nonetheless, the final monitoring systems might produce an unmanageable amount of data which jeopardises all the claimed advantages of the SHM and prevents it from being valuable for the managers. Confusing and low quality information increases the scepticism of the stakeholders regarding the SHM even though the cost of its hardware components continuously decreases, turning them potentially worth [44].

Thus, the real challenge is to join together all the high-level components in order to build up a cost-effective system. This holistic approach includes, in a shared vision, all the system components, aiming at a common goal: an optimised sensing system, avoiding over-instrumentation, which produces a dataset large enough to be properly managed and processed, in order to generate few and clear information with high industrial and societal relevance.

Such a holistic vision needs a proper definition of the SHM paradigm and all the subtasks forming it. In this regard, there exist several attempts to define a general framework for the tasks to be faced in the application of a DI strategy to a specific case study. Farrar et al. [45] developed an SHM program implementation divided into 4 main portions:

- (1) Operational evaluation;
- (2) Data acquisition and cleansing;
- (3) Feature selection;
- (4) Statistical model development.

A modular architecture for the software and the hardware required by a typical SHM system has been proposed by [39], identifying 6 modules:

- (1) Sensory system (SS) module;
- (2) Data acquisition and transmission system (DATS) module;
- (3) Data processing and control system (DPCS) module;
- (4) Structural health evaluation system (SHES) module;
- (5) Structural health data managements system (SHDMS) module;

(6) Inspection and maintenance system (IMS) module.

Modules 1 to 3 form the real-time acquisition and assessment (data-driven approach), whereas modules 4 and 5 form a non-real-time assessment (physics-based behaviour model approach); such division is discussed more in-depth in section 2.1.3. Module 6 stores plans, reports and manuals regarding the design and the maintenance. The subcomponents (software and hardware) of these modules can be rearranged to form only three main categories, irrespective of the dynamic of the assessment (e.g. real or non-real time), as in [46]:

- (1) Sensor system;
- (2) Data processing system;
- (3) Health evaluation system.

According to the data processing stages, Lowe [47] defined a data to decision (D2D) flow, divided in the following steps:

- (1) Sensing: generation of the raw data;
- (2) Pre-processing: data cleansing and reduction;
- (3) Feature extraction;
- (4) Post-processing: feature cleansing, compression, fusion or normalisation;
- (5) Damage Identification;
- (6) Decision making.

As for the present study, a subdivision of the SHM tasks is addressed in a twofold way. First, a framework was developed to split the SHM tasks and support the identification of analogies with the biological systems [48], [49]. The analogies refer to 4 main SHM elements:

- (1) Sensing Unit Architecture;
- (2) Network Architecture;
- (3) Networking;
- (4) Signal processing.

The ultimate goal of this framework was to derive bio-inspired approaches for SHM. Since the present thesis was developed aiming at this global scope, it is deemed necessary to discuss biomimicry in more detail. Thus, section 2.2 is devoted entirely to biomimetics and natural computing.

Afterwards, another framework for the online safety assessment and DI of existing buildings was developed, focusing on the mathematical problems involved:

- (1) Optimisation problem for the design of the sensing network;
- (2) Classification problem for the simple detection of damage;

- (3) Optimisation problem for the validation of the numerical model and the damage localisation and quantification.

In [50], such a framework was presented but limited to the numerical optimisation, to show how a single multi-objective optimisation algorithm could face more sequential problems, from the sensor network design to the diagnosis of the structure, only changing inputs and definition of the objective functions.

All the above frameworks are combined in Figure 2.4 into a single flowchart. It is evident that any DI strategy is composed by a set of sequential steps, mainly divided into two groups:

- (1) Upstream stage (development of the strategy), needed to define the specific problem instance and the tools to face it – namely the operational evaluation portion and the optimal sensor placement task;
- (2) Continuous loop (application of the strategy), composed of all the remaining portions and tasks.

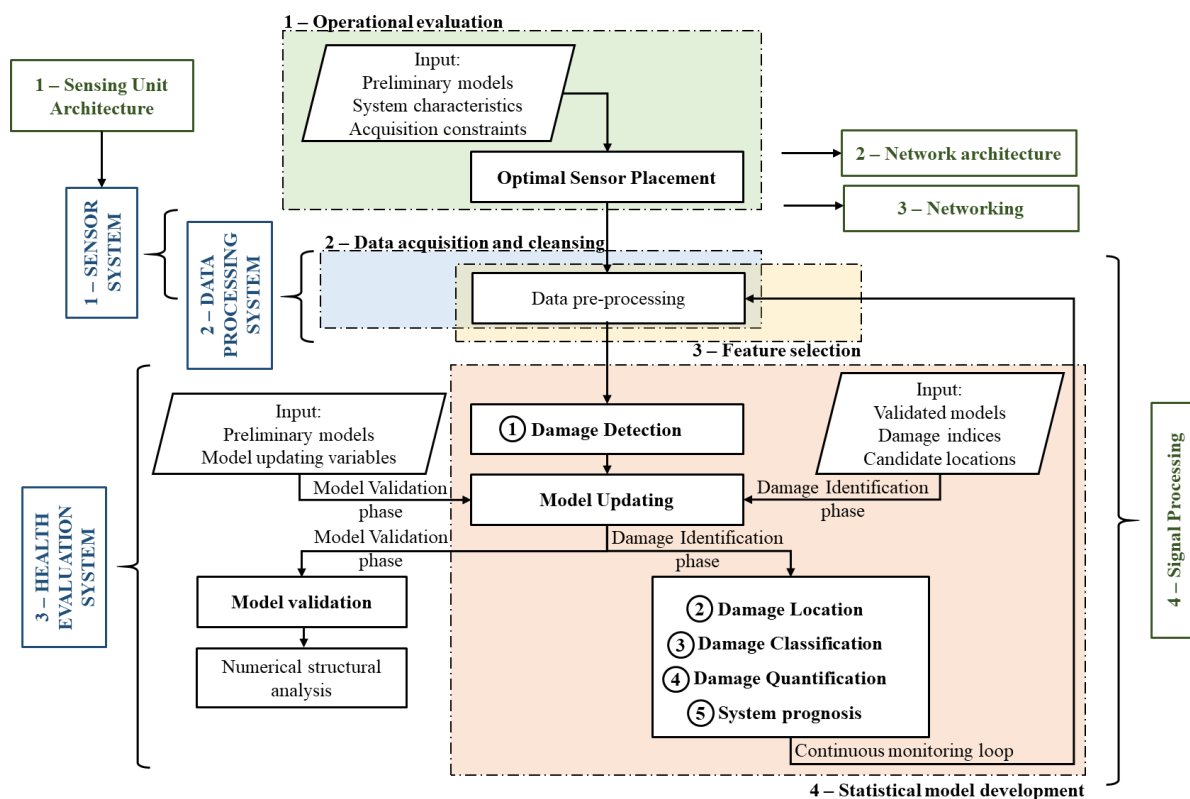


Figure 2.4: General framework for a Damage Identification strategy.

The upstream stage collects the definition of both the problem instance and the monitoring strategy (e.g. characteristics of the system under investigation and its operational and environmental conditions, quantities to be observed and quantities of interest related to the observed ones, type of software and hardware tools involved, constraints to be considered in the deployment, etc.). Usually, a preliminary

physic-based model of the system is required. This stage corresponds to the operational evaluation and the deployment for data acquisition of Farrar's frameworks. Here, it is called for simplicity network design (see section 2.1.1), where all the components to be used, belonging to the modules described by [39] should be identified. The output of this stage is the definition of the network architecture and the networking strategies required (e.g. time synchronization, energy harvesting, connection of the sensors in more layers, etc.), according to Masciotta et al. [48].

The second stage consists of a continuous loop of data acquisition, pre-processing and post-processing. In the framework by Masciotta et al. [48], this loop is called generically signal processing, being subdivided in many specific subtasks. However, two main moments can be identified. The first moment is the data pre-processing (see section 2.1.2), through the sensor and the data processing systems, where the feature extraction is a specific subtask as data acquisition, cleansing, fusion, encoding, compression, storage, etc. (in the flow developed by Lowe [47], these coincide with the first 4 steps). The general term of pre-processing is here used, since, in many cases, not all the steps are taken or not in the same order, and many techniques address more steps at the same time (see section 2.1.2). Then, the second moment is the data post-processing with model development (see section 2.1.3), through the health evaluation system, and the decision making according to the D2D chain. This step consists in the definition of the action to be taken, based on the output of the assessment and a list of feasible actions. Some authors call the model development "statistical". This implies that the QoI are analysed as stochastic variables. Many existing approaches do not rely on a statistical analysis, for instance the neural models or the negative selection algorithms. However, it is a shared opinion that the nature of the QoI is clearly stochastic and that the general framework of the SHM must be supported and cast into statistics. Nonetheless, simple subcomponents of such framework can rely on algorithms and methodologies which do not consider directly the stochastic nature of the analysed processes. The model development addresses the 5 steps of DI in two ways:

- (1) The features are analysed online to verify that the system continues in its normal state (damage detection);
- (2) Whenever an anomaly is detected with confidence, the features are used to identify the damage scenario (damage localisation, quantification and classification) and to predict the future behaviour (prognosis).

The specific characteristics of DI allow a further simplification since a damage state is not naturally reversible, thus the problem is not the identification of a single outlier point, but of a sequence of points [28]. The resulting framework is composed of many subtasks and collects many techniques which have

been developed in different disciplines. However, any successful application to a real-world case study requires a unitary design and implementation of the global strategy through the definition of the best methodology for each subtask and the interconnection among them.

In the following sections, more details are provided on the main component of the DI strategy.

2.1.1. Network design

The network design stage aims at the definition of all the monitoring system components, namely [45]:

- Type of observables to be acquired and features to be extracted;
- Sensors type, number and location;
- Bandwidth, sensitivity and dynamic range;
- Data acquisition/telemetry/storage system;
- Power requirements;
- Sampling intervals (continuous versus periodic monitoring and programmed versus triggered events);
- Processor/memory requirements;
- Excitation source (active sensing).

The quality of the collected information strongly affects the effectiveness of the state assessment. The quality itself depends on the quantities to be measured and on the procedure to measure them [51]. Therefore, given a quantity to measure, the quality of information can be improved by enhancing the sensor technology (e.g. accuracy, resolution, sensitivity, noise, etc.) and/or the sensor network design (e.g. number, orientation and deployment of the sensors) [35], [51]. The improving of the sensor network design is addressed by the so-called optimal sensor placement (OSP) which refers to the definition of the optimal combination of type, number and location of the sensors, under some performance metrics, for a specific case study. The effort spent in this field has been much less and, thus, large room for improvement exists. More in details, the OSP is composed of 8 main steps [35]:

- (1) Definition of the demands;
- (2) Definition of the operational parameters;
- (3) Choice of sensors type;
- (4) Determination of the objective function;
- (5) Choice of the OSP technique;
- (6) Definition of the inputs;
- (7) OSP resolution;

(8) Deployment.

The steps (1) to (6) consist in the definition of all the main aspects to set the specific problem instance and the OSP algorithm instance, whereas, in the last two steps (7) and (8), the OSP is carried out and the network is deployed over the investigated system. Note that, in the list above, points (2) and (3) are inverted with respect to the original list in [35], as it is opinion of the author that the sensor type must be selected based on both the demands and the operational parameters. The definition of the scope of the monitoring – step (1) – is essential to identify the requirements and it mainly consists in the definition of the Qol one wants to infer. Whenever the Qol are not directly measurable, it is necessary to define the observables related to them. Commonly, the Qol are not only affected by the damage but also by other variables, hereafter called sources of uncertainties. A proper definition of these sources helps filter out their effect on the Qol, magnifying the damage dependency.

The set of operational parameters – step (2) – includes the final set up of the qualitative and quantitative problem-specific requirements, which can be [35], [45]:

- Related to the damage: type of damage to be detected (e.g. crack, fatigue, corrosion, loose connection, etc.), length scale of the damage to be detected, time scale of the damage evolution, probable damage locations, thresholds on the acceptable performance, etc;
- Related to the boundary conditions: expected variability range for environmental and operational conditions, effect of the EOVs or of harsh conditions on the monitoring system;
- Related to the budget (economical parameters): initial cost of the network, maintenance costs, upgrading costs, etc;
- Related to the system (topological parameters): size, arrangement, sensor spatial relations, accessibility, adverse effect of the sensors, nontechnical issues, etc;
- Related to the sensing system: energy requirements, available number of sensors, length and deployment of the cables for wired sensors, etc.

The characteristics of the expected damage are essential to tailor the DI strategy to the specific application needs. For instance, the time scale of the damage to be detected limits the sampling rate, whereas the length scale has a direct influence on the excitation source, if required, and on the bandwidth of the instruments, which, in turn, determines the observables and the sensors. The development of a robust damage model is still an open challenge which strongly affects SHM especially for civil engineering. The definition through reliable models of the system damage states can boost the sensing unit architecture development and the network design optimisation through the identification of specific requirements as: expected location and characteristics of the damage and measurement conditions [52]. Furthermore, the

existence of a damage model is essential for physics-based model development, as described in section 2.1.3. In real case applications, such models can be direct (e.g. crack size) or indirect, as the effect of a damage scenario on an observable quantity (e.g. change in the vibration response). However, there is not a single way to model all the possible damage scenarios that might affect a system. The definition of the requirements related to the damage scenarios benefits from experimental campaigns. In Civil Engineering, system-level testing to failure is almost impossible, unless for very limited cases of scaled-down models. Component-level testing is economically more viable. However, engineered components are always only nominally identical, thus, statistical variations and uncertainties should be considered [53]. Most of the experimentation which can be carried out is numerical and, in many real cases, the definition of the expected damages is only based on the experience and intuition of the technicians [54]. Numerical simulations are also essential to estimate how sensitive is a given feature to the damage progression. This is equivalent to an estimation of the derivative of the feature with respect to the damage. The comparison of the damage sensitivity of a feature with the uncertainty in its estimation allows to ensure that variations due to uncertainties do not cover the variation due to damage. However, large and complex systems are likely to experience several different damage mechanisms, but the network optimised for one might not be suitable for the identification of others. A good knowledge regarding the possible damages and a reliable damage model are, thus, essential also to rank the mechanisms based on the probability of occurrence and the severity of the consequences [54]. The boundary conditions refer to operational and environmental conditions under which the monitoring is carried out. This information is twice essential since: on the one side, the data pre-processing must remove the trends in the features due to such conditions and it is eventually recommended that specific sensors to help filter out their effect or to monitor them are added; on the other side, the sensors have limitations regarding the range and the characteristics of the environmental and operational conditions [54]. Operational and environmental ranges, together with the damage characteristics, limit the suitable sensors, data acquisition systems, candidate features and models for the post-processing. Among the set of suitable elements, the network is optimised considering a set of limitations, mainly related to the economical parameters, to the investigated system and to the sensing system, encoded as numerical constraints or penalisation factors into the optimisation problem [35].

The more detailed is the pre-definition of all the above operational parameters, the more effective is the network design. Indeed, whenever the requirements are not properly defined, there is a high risk of developing an SHM system with poor performance, unable to fulfil its objectives. Nonetheless, most of the applications until the last decade were not supported by a proper pre-definition of the requirements

along with a testing stage to validate. Commonly, well-known sensors were sparsely distributed and the features were selected “after the fact” among the available stored data [54].

The definition of demands and requirements lead to the identification of which SHM methodology shall be used. The most common methodologies belong to one of the following 5 classes of techniques [35]:

- Vibration-based monitoring;
- Strain monitoring;
- Elastic waves-based monitoring;
- Electromechanical impedance-based monitoring;
- Comparative vacuum monitoring.

Each of them allows to measure a specific feature and relies on a specific set of sensor types – step (3). In many applications so far, the sensor placement has been carried out only based on engineer experience and intuition, if needed, supported by a preliminary numerical modal analysis for the visualisation of the mode shapes and the consequent selection of meaningful measurement points. The mode shapes, in fact, allow to localise the points where the displacement response is maximum. This approach with no or minimum mathematical formulation cannot be extended to systems with complex local modes or coupling phenomena, thus it leads to an acceptable solution only for global modes which can be easily identified. On the contrary, the OSP is a numerical optimisation problem that can be solved using specific algorithms (for more details on the optimisation problem formulation see section 2.1.3). More in details, given a number of sensors n_{sens} , the OSP problem consists of finding among a set of n_{cand} candidate locations, where $1 \leq n_{sens} \leq n_{cand}$, the n_{sens} locations and orientations which are optimal for a set of given objectives. Although the structure has infinite nodes the candidate locations are a finite set. It is thus a specialised Knapsack problem. A solution to the OSP problem is a vector of spatial nodal coordinates $\mathbf{d} = \{x_1, y_1, z_1, \boldsymbol{\delta}_1, \dots, x_{n_{sens}}, y_{n_{sens}}, z_{n_{sens}}, \boldsymbol{\delta}_{n_{sens}}\}$, where $\boldsymbol{\delta}_i = \{u_i, v_i, w_i\}$ is the orientation of the i th sensor and $\{x_i, y_i, z_i\}$ correspond to its location in the reference system of the geometrical space. A simpler definition is achieved by resorting to a discretization of the investigated system and to the enumeration of the candidate DOFs: $\mathbf{d} = \{d_1, \boldsymbol{\delta}_1, \dots, d_{n_{sens}}, \boldsymbol{\delta}_{n_{sens}}\} \in \mathbf{D}$. The problem formulated above is a constrained combinatorial optimisation, whose solution requires the definition of an objective function f_{obj} – step (4) – namely a criterion which allows to rank the goodness of a candidate solution considering the identified requirements as constraints. The problem is assumed to look for a minimum of the objective function, namely:

$$\text{Find } \mathbf{d}^* = \arg \min_{\mathbf{d} \in \mathbf{D}} f_{obj}(\mathbf{d}) \quad \text{s.t.} \quad g(\mathbf{d}) = n_{sens} \quad (\text{Eq. 2.6})$$

The dimension of $\mathbf{D}$ for a given number of sensors n_{sens} and candidate locations n_{cand} is:

$$|D| = \frac{n_{cand}!}{n_{sens}! (n_{cand} - n_{sens})!} \quad (\text{Eq. 2.7})$$

namely the number of all the possible distinct sensor configurations.

The objective function is not known analytically, thus the OSP is a black box optimisation problem. Indeed, the objective function value in a specific point can be calculated only by running a simulation. This requires the existence of a preliminary numerical model of the investigated system. Usually, a mock-up finite element model is used. The output of the model allows to estimate the objective function for the attempt solution, namely a set of degrees of freedom simulating the deployed sensors. Based on the adopted algorithm, several attempts are tried and compared. Finally, the best solution to the optimisation problem is the argument of the optimum for the objective function. Furthermore, in its most simplified version, the OSP is also a combinatorial problem, since the attempt solution is a discrete variable which can assume only values from the set of the candidate sensor locations. Combinatorial optimisation problems can be solved with a brute-force approach by evaluating all the possible combinations. For large complex systems, this approach becomes quickly infeasible and the objective function is commonly non-convex, presenting many local optima. Therefore, a proper optimisation algorithm must be selected – step (5). The algorithms suitable for a combinatorial discrete constrained black box optimisation problem can be classified as deterministic or stochastic. Among the deterministic ones, a class of optimal optimisation methods exist only for smooth continuous convex objective functions and they require information regarding the gradients or the Hessian of the function, which is almost impossible in real cases. A class of sub-optimal deterministic methods is composed by a group of simple heuristic sequential sensor placement (SSP) algorithms. The solution achieved by the simple heuristics is sub-optimal and, if the objective function has high dimension and it is complex (e.g. non-linear and with many local optima), the performance becomes quickly unacceptable. Thus, OSP can be addressed through stochastic methods. Such class of methods can be subdivided into two:

- Techniques whose solution is achieved through a stochastic process (e.g. the metaheuristic algorithms);
- Techniques that treat the variables as stochastic (e.g. the Bayesian processes).

2.1.2. Data pre-processing

Due to the improvement in sensing architecture (e.g. reduced size and cost, increased reliability and ruggedness), SHM systems can be composed of large quantities of sensors belonging to different typologies and producing data with different sampling rate and formats. The result is a large produced

information potentially able to reduce the uncertainties, but extremely more demanding in terms of data management and interrogation [52]. It is clear that a monitoring system can easily produce terabytes of unstructured or semi-structured data per week [55]. The lack of structure is due to the fact that signals from several type of sensors are managed together and that changes in the loading or the external conditions produce different patterns in the signals. This is further worsened by a series of possible physiologic anomalies in the signal itself. The complexity and the volume of this data flow is comparable to the so-called “Big Data”, which refers to datasets that are almost unmanageable and unprocessable with traditional approaches and tools [37], [44], [56], [57]. The pre-processing stage aims at reducing the complexity by analysing the raw data to produce a reduced set of meaningful features. Typical processes applied in this stage are data cleansing, data reduction, data fusion, data normalisation and feature extraction. Each of the previous tasks can be addressed by stand-alone methodologies or combined with other tasks.

Data cleansing is a discrimination process of the data which can be accepted, corrected or rejected. Correction and rejection are possible strategies to deal with corrupted or inaccurate records. Before the “Big Data” era, such selective approach of discrimination was manual and carried out by experts, thus highly time consuming and not automatic [58]. Automatic methods are developed in [59], [60]. Redundancy through backup sensors allows to substitute a malfunctioning sensor by switching on its backup, thus, the data is reconstructed without any statistical estimation. Clearly, for dense networks, doubling the deployed sensors is impossible. Stronger computing units allowed the development of self-validating wired sensors (SEVA) [61], [62] and of smart wireless sensors (Xnode) [63]. The former assesses online the quality of their own measurements, whereas the latter can self-diagnose, self-calibrate, self-identify and self-adapt. However, in some scenarios, uncertainty calculation and anomaly correction are expertise-dependent, and in general, such sensors are still in a prototype stage [58]. In a broader sense, cleansing addresses irrelevant data as well, which is also the aim of data reduction and data fusion (also called compression or aggregation). Data reduction and data fusion are essential because any learning or optimisation algorithm, which is needed to solve the DI problem, suffers from the curse of dimensionality. Namely, this phenomenon consists in a significant growth of the feature space volume with the dimensionality of the feature itself. In large feature spaces, to avoid sparsity of the data analysed, an unmanageable quantity of samples is required, being these samples, for instance, training data of a learning algorithm or iterative attempt solutions of an optimisation algorithm. Many machine learning strategies allow this transformation from one feature space into another (e.g. principal component analysis, factor analysis and Sammon mapping [32]). Common data compression processes

are decimation (length reduction) and data fusion (dimensionality reduction). Data fusion consists in merging information from different sources, with various conceptual, contextual and typographical representations [64]. The output is a new dataset with same reliability and information content but reduced dimensionality and uncertainty. Thus, data fusion increases the effectiveness. In general terms, data fusion is the process of combining information from different sources to produce a more easy-to-handle information.

Even when the noise has been filtered out from the signal, all the incorrect information has been rejected and the correct one has been reduced to a low dimensional dataset, the response might still be contaminated and variable. Normalisation aims at suppressing the variability in the data due to phenomena which are not relevant to the specific goal of the post-processes [44]. Namely, when the goal is the damage detection, non-damage induced variability is irrelevant or even detrimental. EOVs strongly affect the system behaviour since they influence the mass, the stiffness and the stress level of the system. Typical environmental conditions are [33]: aerodynamic and hydrodynamic loading; radiation environment; temperature, moisture and humidity levels. Operational conditions usually depend on the live-load conditions, which can be static or dynamic (e.g. traffic-induced loading). Civil Engineering systems related to significant mechanical processes (e.g. hosting heavy machinery, power plants, oil extracting etc.) undergo also different conditions due to power fluctuation, start-up and shut-down transients, moving parts manoeuvres, changing mass loading (e.g. fuel levels in tanks, payloads of vehicles operating over the system, etc.) [33]. Many evidences of the effect of EOVs on the structural dynamic response have been discussed in the literature [65]–[70]. Long-term monitoring of monuments confirmed the influence of environmental conditions also for this type of structures [71]–[80]. The induced variations are extremely quick, presenting non-negligible shifts in the seasonal plots of the features; moreover, their effect can easily mask the shifts in the system properties due to low-to-medium damage extents [81], [82]. When the measurements of the ambient factors to be filtered out are not available, normalisation is effective only if no correlation exists between such factors and the damage effects. On the contrary, whenever the environmental and damage effects induce similar consequences, a discrimination in the absence of measurements is more difficult. In any case, the baseline data must be as much representative as possible of all the expected EOVs ranges. Examples of normalisation strategies are discussed in [33], [66], [83]–[87].

Damage-sensitive features, thereafter referred for simplicity just as features, are multivariate outputs collected into a vector and directly correlated with damage. Feature extraction is the process of transforming the observables into the QoI representative of the system state, thus it is a transformation

from the measurement space into the feature space. It is called feature selection a further data reduction which aims at discarding the elements less damage sensitive. Automatic feature extraction strategies have been developed to correlate pairs of observables and features irrespective of the investigated problem. Some extraction strategies, instead, are problem-specific and require that the damage sensitivity of the features is verified.

In conclusion, it is worth noting that all the pre-processing strategies that are described here and in the previous section change the dimensionality of the input space. Even though this is essential to turn the problem manageable and even though most of the dimensions carry few or no information regarding the damage, the reduction must be done carefully to assure that the dimensionality of the feature space is reduced preserving the quantity of information carried. Namely, the reduction is effective only if the damage sensitivity is not projected out along with the sensitivity to other sources of uncertainties. However, it is known that the reduction increases the damage sensitivity in some direction of the original space but decreases it in others. An example adapted from [28] clarifies the issue. Imagine that the samples of the same two-dimensional feature space of Figure 2.2 are projected onto a unidimensional space corresponding to the narrow direction and let the thresholds for damage detection be defined based on the confidence interval (dashed lines in Figure 2.5). In the original two-dimensional space, the decision boundary was the ellipse, thus the damaged sample, in red, was easily distinguishable. Due to its position, such point is instead labelled as normal in the reduced space.

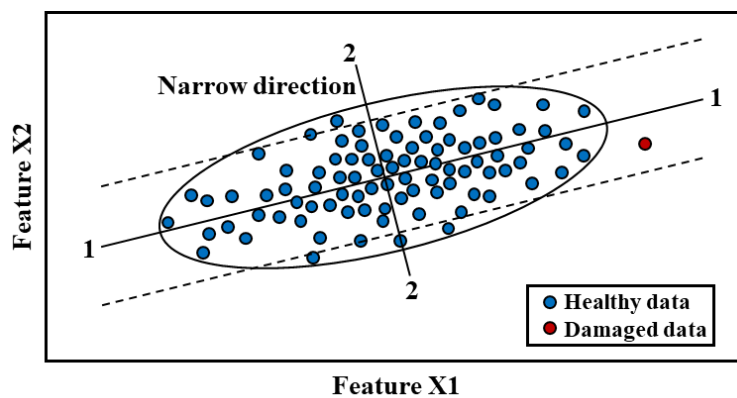


Figure 2.5: Bivariate data samples projected onto the narrow direction axes (modified from [28]).

In conclusion, pre-processing the data is essential for the performance of the damage detection. Hence, it is stressed one more time that the definition of the processes required in this stage and their setting should be done in a holistic approach, namely having clear in mind the characteristics of the system and the strategies used to develop the statistical model and perform the DI. This approach drives a well-

thought-out dimensionality reduction by transforming the observables into a reduced set of Qol highly damage sensitive and makes the DI methodology cost-effective, thus more attractive to stakeholders who might invest in it.

2.1.3. Model development

This stage of the DI strategy consists in the post-processing data interrogation, namely the development of a numerical model that is able to distinguish statistically significant changes in the Qol [45]. It is assumed that the system damage which is to be identified affects the Qol to such an extent that the damage can be characterised [88]. Whenever a new record of the Qol is significantly different from the normal state, the measurement is classified as an anomaly. The assumption in the SHM theory is that, once any other effect able to disturb the Qol value is filtered out in the feature extraction stage, the anomaly is a symptom of a damaged condition (Axioms IVa and IVb). Good choices of the features simplify the identification to such an extent that the simple visual inspection of the changes in their trend allows the anomaly detection. However, numerical approaches are required in order to identify the damage in the earliest stage and to deal with the uncertainties in the features. The 5 DI levels can be addressed with different numerical techniques belonging to two main families:

- Model-based algorithms;
- Data-driven algorithms.

Mathematically, the former group belongs to mathematical optimisation, whereas the latter to machine learning. The main difference between the two is clarified by the black box model of computer systems which identifies three main elements [89]:

- (1) The input;
- (2) The model;
- (3) The output.

In optimisation problems, the model and the output are the known parts while the unknown part consists of the input that, computed using the model, gives the desired output (inverse problem). Machine learning, instead, is a modelling problem since the input and the output are known, whereas the model is required and it is inferred or learned (forward problem) [89], [90]. The model is not parametric and not fixed, quite the reverse, it can change as result of the learning process and the data available. In Figure 2.6, the flowchart of both approaches, data-driven and model-based, is presented, whereas in Figure 2.7 a taxonomy of statistical model development is shown, summarising the information that are further discussed in the present section. In the blocks close to the two main families, the Rytter's hierarchy is

reported, highlighting in green the levels that can be easily reached by the techniques and in red the ones that are hardly achieved and deserve more research.

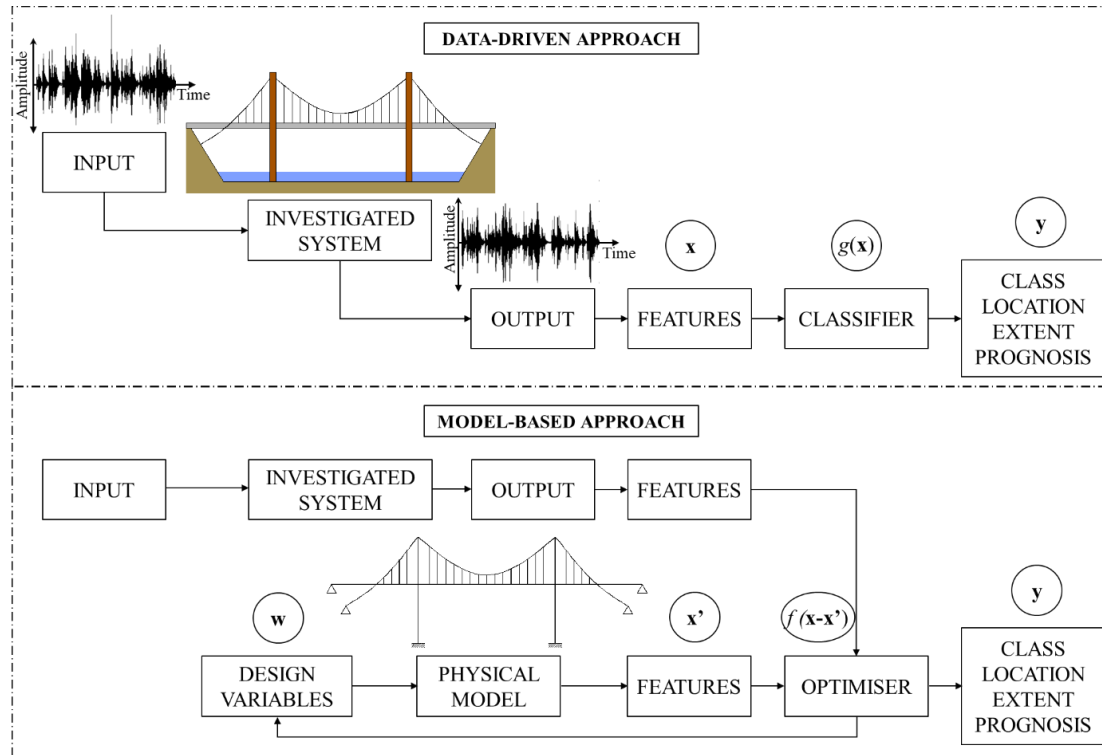


Figure 2.6: Flowchart of the data-driven and the model-based approaches.

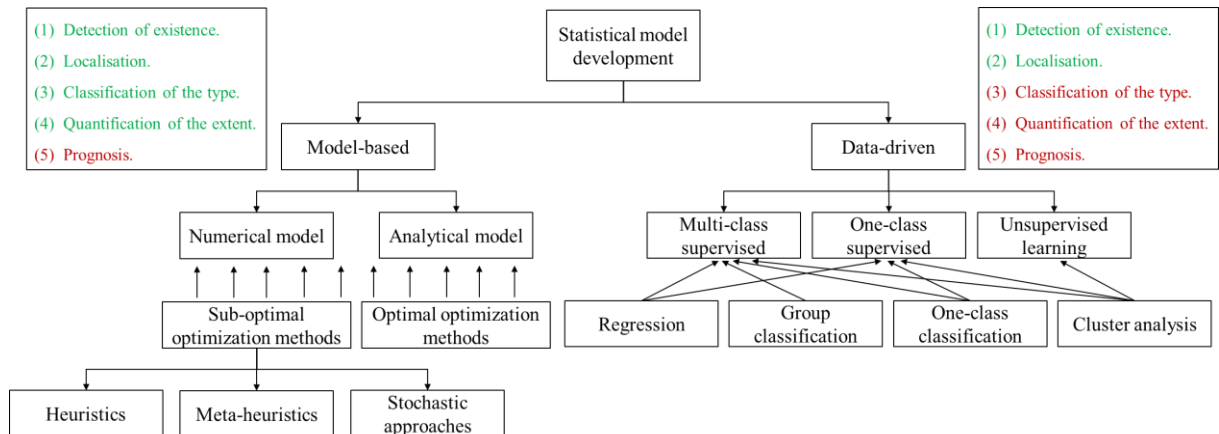


Figure 2.7: Taxonomy of statistical model development divided into the two main approaches and the possible characteristics of the problem. The Rytter's hierarchy with the achievable DI goals and the viable algorithm types for each problem are presented.

Model-based methods are so-called since they address DI by model updating. A physics-based or law-based numerical or analytical model of the system is required (e.g. based on the Finite Element Method, FEM, on the Finite Difference Method, FDM, on the Boundary Element Method, BEM) and physical properties are tuned by matching the experimental measured data and the model output. Mathematically,

this is a continuous constrained black-box optimisation. Mathematical optimisation is a branch of mathematics and computer science whose main objective is the definition of the best element from a set, under boundary conditions and constraints. As in the case of OSP, the optimisation requires the definition of an objective (or cost) function. Since minimising the objective function f_{obj} is the same as maximising $-f_{obj}$ [91], without loss of generality, it is possible to define the optimisation problem as the following minimisation:

$$\min_{\mathbf{w} \in \Omega} f_{obj,i}(\mathbf{w}), \quad (i = 1, 2, \dots, M) \quad (\text{Eq. 2.8})$$

subjected to: $h_j(\mathbf{w}) = 0, \quad (j = 1, 2, \dots, J)$

$$g_k(\mathbf{w}) \leq 0, \quad (k = 1, 2, \dots, K)$$

where $f_{obj,i}(\mathbf{w}): D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+$ is the i -th objective function in the problem; $D_1 \times \dots \times D_n$ is the search space given by the variable domain $\Omega \subseteq \mathbb{R}^n$, while $h_j(\mathbf{w})$ and $g_k(\mathbf{w})$ are constraint functions, being the constraints a binary evaluation regarding a specific requirement that the solution must satisfy. All the functions depend on the design vector $\mathbf{w} = (w_1, w_2, \dots, w_n)^T \in \Omega$ whose components, w_i , are called design or decision variables. Such variables can be continuous, discrete or a mixture of both. Ω is the region of feasible solutions in the search space. A feasible solution will be $\mathbf{s} = \{(w_1, v_1), \dots, (w_n, v_n)\}$ where $v_i \in D_i$ and $\mathbf{s}$ satisfies all the boundary constraints [92]:

$$\mathbf{s} = \arg \min_{D_1 \times \dots \times D_n} \left(\min_{\mathbf{w} \in \Omega} f_{obj,i}(\mathbf{w}) \right), \quad (i = 1, 2, \dots, M) \quad (\text{Eq. 2.9})$$

If $M = 1$ the problem is called single objective, whereas if $M > 1$ is called multi-objective. The best solution to the optimisation problem is the argument of the optimum for the objective function. In vibration-based SHM, the objective function is commonly dependent on the difference between numerical and experimental features $f_{obj}(\mathbf{x} - \mathbf{x}')$. Based on the design value $\mathbf{w}$ iteratively updated, the features are calculated and compared to the experimental ones. The process continues until an acceptable minimum of the objective function is found. The best solution, in fact, is the set of physical properties whose calculated features minimise the objective function. The problem is black box since the objective function is known only by running a simulation in the numerical model. Model-based methods potentially allow to address the DI up to the last level (prognosis). A basic assumption is that the changes in the parameters of the model are related to damage-induced changes in the system element associated with the parameters [88]. Thus, the changes in the model characteristics allow to locate and to quantify the extent of the damage. However, the approach suffers of several shortcomings, due to the sources of uncertainties which affect the definition of the numerical model (modelling assumptions as boundary conditions, material properties, geometrical discretisation, etc.), and the ones associated with model

prediction. Furthermore, sources of uncertainties other than damage affect the Qol, rising the problem of non-unique identification. Indeed, reducing the amount of available data, the number of parameters which can be uniquely identified decreases [88] and, in general, due to compensating errors, many model predictions might match the observations during the calibration even if they are unrelated to the actual damage scenario [93]. Model-based approaches also require higher expertise and are computationally more expensive. Finally, since the objective function is not analytically defined but by realisations (e.g. analysing the model for the specific nominal values of the parameters), it is not possible to define *a-priori* the sensitivity of the objective function to the model parameters change. The shape of the objective function cannot be easily predicted and this can be multi-modal (e.g. with many peaks) or flat (e.g. low sensitive) [88].

Data-driven approaches reduce the sources of uncertainties but have limited applicability to the higher levels of DI (e.g. higher than localisation) and they are strongly affected by the measurement procedure (e.g. number and location of the measurement points and frequency content of the excitation). Furthermore, the definition of a reliable non-physical model usually requires large number of records in many different normal conditions. Data-driven approaches aim at the development of a numerical non-physical model of the system features. No physical properties are directly analysed, and the process only relies on the experimental output over time. Mathematically, this is a machine learning problem. Machine learning is a subfield of computer science whose objective is the development of methods to identify data patterns and support prediction or decision [94]. The name specifies the main purpose of the discipline, namely giving to computers the ability to learn without being explicitly programmed. By training on a large amount of known data, machine learning methods extract a suitable approximation of the correlation between input and output, as human beings are able to make prediction or decision, based on experience and recognizing analogous situations [95]. A widely used subdivision identifies two types of learning:

- Supervised;
- Unsupervised.

The main difference lies in the available knowledge about the system. Higher level DI can be achieved through supervised learning, which requires a supervisor, namely a set of available examples of the Qol for any of the expected classes. This can be produced by physics-based models (e.g. FEM, FDM, BEM) or experiments, extending the preliminary tests used for damage simulation and feature selection. The former are more viable in terms of costs but can be computationally expensive to run and validate [54] or even under representative for complex geometries and/or complex materials, where heavy refined models are needed or constitutive laws are inaccurate [1]. On the other side, the experiments require the

production of several copies or reproductions of the system to reproduce different expected damage scenarios. In real-world applications, it is extremely common that no knowledge exists about different structural conditions over time. For instance, the training data samples can belong only to the reference, healthy state. In this case, the supervisor is “limited”. Otherwise, in the worst case, a completely unsupervised learning case is given when the label of the actual state is not known, namely the supervisor does not exist. This happens when the actual state is already damaged and, in general, when no baseline is available (e.g. dealing with an old existing building never monitored before or in post-event situations). This approach can address DI only for the levels 1 and 2.

Irrespective of the supervisor, machine learning methods identify damages as emerging data patterns with relevant differences with respect to normal values. Such a problem belongs to pattern recognition, a branch of machine learning. Four main families of “pattern classification” or “pattern recognition” processes for data-driven classification exist [26], [44]:

- Regression;
- Group classification;
- Outlier analysis;
- Cluster analysis.

Regression algorithms are significantly different from the others in the list. The goal of the regression analysis is to define a correlation between specific features (predictors) and particular locations or extents of damage (dependent variables). The features are mapped to a continuous parameter (e.g. a spatial location or a remaining-useful-life temporal parameter), thus the output is not a discrete classification [26]. Regression algorithms demonstrated to be effective for level 1 DI [96], [97] but also applications up to level 4 have been attempted [90]. The other approaches can be framed in the field of classification methods and clustering methods. Among them, group classification is only suitable for binary or higher-level classification through supervised learning, whereas outlier analysis is also suitable for a “limited” supervisor. Finally, the cluster analysis allows a completely unsupervised learning.

Group classification aims at defining boundaries in the feature space between samples belonging to different conditions according to the known labels [44]. As many classes as the known labels can be identified, accounting for different damage scenarios. New samples are then placed in the feature space and labelled according to the label of the class to which boundaries they belong. Many approaches are suitable for this task as neural networks, multi-class support vector machines, Bayesian classifiers and decision trees [44], [98], [99]. Mathematically, the classification for level 1 DI, in a supervised learning way, is called two-class or binary classification [100], being two the system conditions. Recalling (Eq. 2.2)

and (Eq. 2.5), the binary classification aim is to find a mapping function $g(\mathbf{x})$ to model $f^{-1}(\mathbf{x})$ such that:

$$y = g(\mathbf{x}) \quad (\text{Eq. 2.10})$$

where $\mathbf{x} \in \mathbf{X}$ is the vector of the measured system features and $y \in Y = \{Nonself, Self\}$. $\mathbf{X}$ is the set of the complete population collecting all the possible samples. The mapping function or classifier $g(\mathbf{x})$ labels each new measured feature as self or nonself, viz. as belonging to a normal or an abnormal state. The function g is produced by a learning algorithm A based on a training set R of data samples whose label is known. The development of the classifier is thus: $A(R) \rightarrow g$.

As said before, for real-world applications it is more attractive – and feasible – to carry out the classification for level 1 DI in a “limited” supervised learning way. This problem is called anomaly detection or one-class classification and the training data belong only to one of the two classes, viz. to the normal state. It is worth mentioning that as stressed by [44], this is not an unsupervised learning process itself, since known pairs $\langle \mathbf{x}, f^{-1}(\mathbf{x}) \rangle$ exist even if only for the healthy examples. The suitable methods for this problem aim at defining boundaries around the known samples. Thus, new samples are placed in the feature space and can be only classified as inliers or outliers. Several statistical approaches are suitable for the outlier analysis. The goal is to learn and identify the distribution of the training samples, all belonging to the self distribution, and to test how likely is a new sample to belong to it or to a new distribution that is damage induced [26]. Density estimation is the most complex problem in the machine learning discipline which can be addressed in a more general way through nonparametric approaches (e.g. the kernel density estimation). Parametric approaches, instead, need more assumptions regarding the form of the density function, reducing the generality, but suffer less from curse of dimensionality and require the estimation of only few parameters [1]. Among the techniques of this second approach, it is common to assume a Gaussian distribution. Deviation statistics test are commonly used for univariate and multivariate outlier analysis. The discordant values in univariate statistics protrude from one or the other end of the data set [101]:

$$z_{\zeta} = \frac{|x_{\xi} - \bar{x}|}{s} \quad (\text{Eq. 2.11})$$

where the potential outlier x_{ξ} is compared to the samples mean and standardised, dividing by the sample standard deviation. The definition of mean and standard deviation can be inclusive or exclusive whether the potential outlier itself appears in the estimation or not. Usually the inclusive measurement is typical of the offline monitoring, whereas exclusive is preferred for online monitoring, since each new record is analysed as a potential outlier. The discordance z_{ζ} is compared to a threshold. A multivariate discordance value is the Mahalanobis squared distance:

$$D_{\zeta} = (\mathbf{x}_{\xi} - \bar{\mathbf{x}})^T \mathbf{s}^{-1} (\mathbf{x}_{\xi} - \bar{\mathbf{x}}) \quad (\text{Eq. 2.12})$$

where the measurement vectors and the covariance matrix are used. A multivariate data set consists of n observations in p variables, namely n points in p dimensional object space. The outliers are more difficult to find in a multivariate distribution, being easily hidden in the data mass [101]. More information regarding the specific problem allows to formulate different assumptions regarding the form. Minimum volume ellipsoid and minimum covariance determinant are two robust multivariate probability density function estimation strategies [44]. As long as the distribution of the self samples in the feature space looks elliptical and unimodal, namely with a single peak, the Gaussian approximation is likely to work; however, in some cases, the distributions are not convex, multimodal or even separated in unconnected regions [1]. In this case, a non-parametric density estimation is needed. Nevertheless, it is effective just for low dimensional feature spaces, due to the curse of dimensionality and the high computational cost. Bio-inspired soft computing methods, as the Negative Selection Algorithms analysed in the present thesis, proved to be suitable to deal with not Gaussian-like distributions. More details in this regard are provided in section 2.2.2 as well as in the next chapter. Another viable method consists of one-class support vector machine [102], [103]. A different approach is based on the identification of trends in the data, through statistical process control. Several different control charts can be generated to analyse directly the raw signal. When this is autocorrelated, it is possible to avoid systematic patterns in the data by using the residuals between the predictive model (AR, ARX, ARMA) and the actual measurement [54]. This method and the density estimation-based approaches carry out the classification according to the hypothesis testing framework, considering significant shift in a parameter distribution [28] or checking whether a new sample is likely to belong to the identified undamaged distribution [44], respectively. Unfortunately, since hypothesis testing might require the description of a probability distribution for the classification output, it strongly suffers from the curse of dimensionality [26]. In some cases, it is legitimate to assume the characteristics of the classification output distribution to move directly to the test statistics, for instance, by applying a Pearson χ^2 -test or Student t -test. In other cases, it is instead possible to rely on non-parametric statistics, which are distribution-free or do not require the definition of the distribution parameters, as the Kolmogorov-Smirnov test or the Wilcoxon-Mann-Whitney test. Examples of the application of the test statistics for outlier analysis can be found in [44].

In supervised and “limited” supervised learning, it is common that the parameter distributions overlap. This situation complicates the definition of the decision boundary. A good strategy to define is based on Monte Carlo simulation. Identifying the boundary for many datasets, the threshold can be set by fitting extreme value statistics (as Gumbel or Weibull distributions) to the maxima and the minima of each

dataset. Threshold setting can be cast in the general parameter setting framework, by using experimental algorithm testing.

Finally, the main goal of the completely unsupervised learning, i.e. the cluster analysis, is to identify subsets or clusters of the analysed dataset with a minimum within-cluster variation and a maximum between-cluster variation [44]. Namely, this is equal to identifying clusters which are inherently homogeneous and dissimilar among each other. The exact condition corresponding to each cluster is unknown. Thus, some suitable dissimilarity measure is required. Several unsupervised clustering techniques exist and are suitable for the damage detection. Among them the Hierarchy-Divisive clustering (HD) and the Dynamic Clouds (DC) methods. Usually, these algorithms are iterative, for instance analysing successive top-down divisions of the dataset. Different approaches are suitable for clustering [44]:

- Partitioning algorithms, as the k-means, k-medians and c-means;
- Hierarchical agglomerative or divisive methods;
- Density-based methods, as DBSCAN and OPTICS.

For a detailed discussion of the applications, the reader is referred to [104]–[106]. Potentially, such algorithms can identify several different clusters in a dataset, however, they can be applied also to the outlier analysis for damage detection. In this case, their job is further facilitated because it is known that in the training dataset only one cluster exists, thus the algorithms are used to trigger an alarm as soon as the second cluster emerges for the first time.

In Computer Science, nature-inspired strategies have been allowing a significant improvement to the algorithms design, both in optimisation and in machine learning, according to a biomimetic approach and leading to a field of research called natural computing. Since the present thesis aims at the application of bio-inspired algorithms to DI, the notion of biomimicry is discussed in the next sections together with an analysis of the possible advantages for SHM.

2.2. Biomimetics and natural computing

Biomimetics is a term coined by Otto Schmitt in the late '50s of the last century which refers to the use of natural sources of inspiration to face engineering and physical problems. It is commonly used to address the nature-inspired design [107]. A list of biologically inspired designs with already developed applications can be found in [108]. Looking for examples in nature is not a recent approach and it is the obvious consequence of the identification of reliable and effective objects and processes, as a result of

billions of years of evolution and optimisation of materials, morphologies and properties, both physical and chemical, in a multi-level organized structure from macro to nano-scale [108].

In SHM, several examples of bio-inspired products exist [109]–[112]. However, such applications are sparse, non-systematic and limited to processes developed in other areas, for reasons other than SHM, and simply applied to this field to address specific subtasks. A systematic and holistic approach which identifies first the SHM requirements and then tackles the existing open issues by looking for successful paradigms in nature is still missing. This relevant lack has been properly stressed by Masciotta et al. [48]. The main goal of the authors was to path the way for the definition of a dialectic framework to support the identification of new bio-inspired paradigms in the context of SHM. The biomimetic approaches are tailored directly to SHM requirements by finding analogous features between biological and SHM systems and by distinguishing between the natural processes that are viable and non-viable for the mimicking. In this regard, a simple tool called compatibility matrix was developed by the authors. In this matrix, the SHM tasks are distributed along the top and the correlated biological features along the lateral side. Table 2.1 reports an excerpt from the aforementioned matrix, limited to a single source of inspiration, namely the immune system from which negative selection algorithms – main focus of the present thesis – also draw inspiration. The matrix matches physiological and engineering processes showing the compatible mechanisms which can be translated into biomimetic systems together with the few which have already found realization.

Indeed, the work by Masciotta et al. [48] has been developed to explore how to tailor bio-inspired concepts to the specific SHM purposes. To this end, special attention has been paid to natural computing algorithms and their suitability for the subcomponents of DI strategies.

The direct application of biomimetic approaches to numerical computing led to the so-called Natural Computing, namely a multidisciplinary field of research, branch of Computer Science, which is concerned with three main objectives [113]:

- Drawing inspiration from natural phenomena to develop new models and problem-solving techniques;
- Drawing inspiration from natural phenomena to develop new materials and computational technologies;
- Using computational technologies to understand and reproduce natural phenomena.

Table 2.1: Compatibility-matrix of functional analogies between biological and SHM systems (cells with dots indicate matching features), modified from [48].

			SHM system's features																								
			Signal Processing					Networking					Sensing Unit Architecture					Network Architecture									
			Real-time feature extraction	Signal encoding	Lossless data compression	Parallel & distributed computing	Noise reduction	Filtering env./operat. variations	Damage detection	Data fusion	Multi-layer sensor interaction	Mobile network supervision	Real-time data transmission	Time synchronization	Energy harvesting	Self-adaptability/organization	Homeostasis	Self-diagnostics & -repair	Learning/Pattern recognition	Threshold detection	Self-activation/shutdown	Power efficiency	Hierarchical topology	Multilevel clustering	Sensor placement optimisation	Real-time decision making	Flexible management
Biological features		Biological system																								Biomimetic system	
IMMUNE SYSTEM	Multilevel defence organisation	Skin-Inn.IS-Adap.IS					●							●								●		●	●		
	Alarm issue	Innate immune system						●										●									
	Pattern recognition	Lymphocytes						●									●									Pattern recognition algorithms	
	Cells communication	Cell receptors							●																		
	Cells clonation	B lymphocytes							●								●									Clonal selection algorithms	
	Cells patrolling	B-cells								●																Software-embedded agents	
	Parallel and coordinated actions	Lymphocytes			●				●			●															
	Pathogen counteraction	Antibodies															●										
	Selection of effector cells	Lymphocytes																						●			
	Self/non-self discrimination	T lymphocytes					●	●																		Negative selection algorithms	
	Cells activation/suppression	T-cells																	●	●							
	Antigen specificity	T-cells						●																		Negative selection algorithms	
	Equilibrium restoration	Lymphocytes													●												
	Immunological memory	Memory cells															●										

These three branches are strictly interconnected, since the implementation of new models often requires an analytical explanation of the phenomenon and both can benefit from the use of alternative devices inspired by nature. Despite its recent definition, the intrinsic aim of Natural Computing was already clear to the fathers of Computer Science, as some of Turing's works demonstrate [114], as well as the first bio-inspired models due to Ulam and vonNeumann dating back to the 1940s. De Castro [113] defined the approach to Natural Computing as “think naturally about computation” and “think computationally

about nature". Mimicking a process requires the identification of its basic mechanisms, usually in a really simplified qualitative rather than quantitative way. This allows a clear identification of the tools required to accomplish the emulation and eventually to develop new proper ones [114]. Natural computing systems try to reproduce specific issues, essential in nature for the surviving of the natural systems. Speed of response and generality of response are particularly relevant. In nature, in fact, it is essential to respond in real-time, namely in fractions of second, to an external input. Computing processes mimic this by introducing high-speed operations, reducing the number of steps required from input to output and producing progressive approximations of the final answer. This is an artificial version of the pre-alarm, namely the use of premature results, typical of living being. The generality is the ability to provide good results against a wide range of inputs [115], [116]. Nature is full of examples of processes whose application in computer science can lead to improvement in computation, even if they are not strictly finalized to a computational aim (in Turing's classical view) but to other purposes as surviving, evolving, seeking. The challenge is to identify them and synthesize them properly [117]. However, some authors (e.g. [118]) reported a likely excessive enthusiasm, which followed the impressive results of natural computing. Several difficulties in the mimicking process must be taken into account. Among those, one of the main is due to the extremely huge amount of time, agents and resources available in natural processes. The high redundancy, the possibility of killing the unfits, the extreme fuzziness and the degree of non-determinism are features hard to implement and manage in computations. Thus, nature-inspired solutions are not always the most straightforward or efficient and only a specific analysis of the problem faced allows the identification of the best approach [113]. Concluding, natural computing is an emerging field which is providing powerful algorithms for some of the most relevant problems in numerical computing (e.g. optimisation and machine learning). The present thesis focuses on one of these algorithm families, i.e. the Negative Selection Algorithms (NSAs) which belong to the so called Artificial Immune System methodologies. In the next section, the Immune System paradigm is described both from the biological and the computational point of view, to identify the suitable processes for SHM goals.

2.2.1. Biological and artificial immune system

The Biological Immune System (BIS) is the body multi-layered line of defence which allows to keep the rate of occurrence of diseases extremely low, considering its continuous exposition to potentially aggressive microorganisms. The BIS is composed by a set of different biological structures (tissues, organs, single cells), and processes, whose aim is to produce the immune response, in order to protect the organism. A detailed overview can be found in many textbooks, as Janeway jr. et al. [119], main

reference for the following part. Three different levels can be recognized within the BIS. A first fixed defence consists of the external and internal epithelial surfaces of the body where the foreign agents firstly assault the organism. Most of the infections are prevented here, by these barriers. At the subsequent level, the Innate Immune System (iIS) repels, without any symptom of disease, a wide range of the pathogens able to overcome the first layer, through a set of specific agents (e.g. macrophages, dendritic cells, mast cells, granulocytes, natural killer cells, innate lymphocytes), always available, since the born of the individual. Seldom, aggressive entities resist, overwhelming the innate response. Just in this case the evidences of the disease appear and the last level of the defence, namely the Adaptive Immune System (aIS), is activated. According to the self/nonself discrimination theory, all the entities, to which immune system agents could be exposed, are subdivided into self or nonself. Obviously, self entities are part of the same organism of the immune system, whereas the nonself entities are potential disease-causing agents (e.g. viruses, bacteria, parasites, fungi), whose scientific name is pathogens. The aim of the BIS is to distinguish self from nonself and to tackle the latter through the immune response (innate and adaptive). All the involved agents have a specific purpose and obey to specific rules, in a decentralized system, thus any central management is not required. Innate response is triggered when cells' pattern recognition receptors identify the pathogens, by bonding to their specific molecular patterns; then other cells, circulating in the blood, are recruited and migrate to the site of inflammation. When the Innate System is not sufficient to eliminate or recognize the pathogens, it triggers, by means of chemical stimulation, the Adaptive System to produce antibodies reacting to the nonself entities, through the antigens. Each antigen is a specific protein structure on the surface of the pathogens which stimulates a specific antibody. The iIS cells continue to cooperate with the aIS to remove the targeted pathogens and, finally, to declare the aggression defeated. Lymphocytes (divided in B and T lymphocytes) are essential for the adaptive response. Each lymphocyte bears, on its surface, a unique variant of a prototype antigen receptor able to bind with the molecular pattern of the antigen, and the strength of the bond, that depends on the complementarity of the structures, is called affinity. Such bond does not have to be perfect. This imperfect detection allows a single receptor to identify a larger class of antigens, by recognising specific patterns of their structure rather than the whole exact structure, otherwise a single receptor could identify only a single antigen.

T lymphocytes during their maturation undergo two relevant processes called positive and negative selection. Positive selection aims at the proliferation of the T-cells which bind to specific disease antigens. However, since cells are not cognitive entities, the body produces also elements aggressive to itself. Negative selection acts on the T-cells produced by the positive selection to prevent autoimmunity, by

killing the cells that bind to self elements. The correct execution of this process leads to an essential property of the immune system, the so called self-nonself discrimination.

The evident similarities between the BIS features mentioned above and numerical optimisation or machine learning classification problems motivated the translation into computer science of the biological processes. It is called Artificial Immune System (AIS) the abstraction of one or more immunological processes to address mathematical problems [120], aiming to reproduce especially the following features [121], [122]:

- Uniqueness;
- Decentralization and distributed detection;
- Communication and cooperation;
- Novelty detection;
- Imperfect detection and generalization;
- Malfunctioning protection against autoimmune attacks;
- Learning and memory;
- Specificity, disposability and diversity.

Each individual is unique and has a unique immune system based on his personal development. In many applications, the system to be monitored exhibits several similarities with a class of items (e.g. in the case of structural systems, a specific building can belong to a given building typology) but its ageing process, vulnerability and exposition can be peculiar. A centralized control is not required, and the immune system agents can be highly distributed in different areas, although communication among them allows the cooperation and a higher effectiveness of the response. The novelty detection property guarantees that the system is able to react also to new pathogens, not only to the known ones. This is possible because the matching does not require a perfect identification; on the contrary, the same element of the BIS is able to detect more pathogens. The imperfect detection allows the flexibility of the system and reduces the number of elements needed by the human body (10^8 - 10^{12}) compared to the possible nonself patterns (10^{16}). The negative selection is a censoring mechanism that prevents the autoimmune attack and turns the BIS capable of distinguishing between self and nonself elements. The structure of any pathogen matched by the immune system is learnt and memorized, promoting the uniqueness of the system itself based on its history. This mechanism makes the response faster and more effective in case of a new invasion from the same pathogen. Memory cells preserve the diversity of the elements that is needed since different cells have different tasks with high specificity and different sensitivity to the pathogens, but, at the same time, no single component is essential (disposability).

In natural computing, the behaviour of the natural immune system is not copied blindly, but rather imitated by introducing some changes to tailor the process to the new computation environment [121]. The self/nonself discrimination theory is a great simplification, indeed, which is largely applied in computer science. It can be considered a partially outdated theory, as the less classical approach to immunology overcame the *a-priori* dichotomy by suggesting that an item is not dangerous *per se* but as a consequence of the interaction between the external impact and the current state of the immune system, which is strongly adaptive [123]. Nonetheless, self/nonself discrimination theory proved to be suitable for the development of effective numerical strategies in many fields of research. The Negative Selection Algorithms (NSAs) analysed in the present thesis constitute one of the main examples.

2.2.2. Nature-inspired SHM

Nature-inspired SHM and optimisation

Optimisation for Civil Engineering applications is usually time-consuming, since most of the objective functions are black box and simulations, eventually through FE or other numerical models, are required to estimate the punctual values of these functions. Metaheuristic algorithms are suitable nature-inspired approaches for black box optimisation which mimic natural processes to search iteratively for the optimal solution. All of them follow a similar pattern:

- (1) Definition of an initial attempt solution;
- (2) Estimation of the objective function of the initial solution;
- (3) Updating of the initial solution to explore the search space;
- (4) Identification of the best solution according to the termination criteria.

Metaheuristics can deal with any kind of objective functions, irrespective of properties as differentiability or continuity, and any kind of variables which can be continuous, discrete or mixed. In general, any knowledge regarding the objective function or its derivatives is not needed.

Most of the methods are population-based, meaning that not a single solution but a set of them is evaluated at any iteration. Population-based metaheuristics can explore in the same run all the feasible domain, being suitable for multi-modal and multi-objective optimisation. The solution updating process presents a deterministic component (based on the best solution achieved in the previous iterations) and a random component. These two components can be formulated in several ways and determine the balance between exploration (global search of the optima of the function all over its domain) and exploitation (local search of the optima around the best solution achieved in the previous steps). The initialization can be deterministic, random or a combination of both, aiming in any case at maximizing

the differentiation of the initial population of solutions. Irrespective of that, the methods are global, thus their accuracy does not depend on the initial population of solutions. Despite their evident advantages, they are usually simpler to implement; however, they also suffer from some shortcomings, usually addressed by developing specific sub-strategies within the global algorithm [35]. First, these methods are sub-optimal. There is no guarantee that the solution achieved is the optimal one as well as there is no guarantee of the convergence. Therefore, there is a risk of local optima trapping and premature convergence, when the exploitation component overcomes the exploration component; whereas, there is the risk of loss of best solution found, when the exploration component overcomes the exploitation component. Basic formulations need usually long time for convergence. The first problem is intrinsic of this type of algorithms; thus, it is convenient to repeat the analysis with different initializations, assuming the best overall solution. It is also worth noticing that a sub-optimal solution can be enough for the type of problem investigated. The following three problems are due to the complicate balance between global and local search of the space. This balance depends on the tuning between the deterministic component toward the best solution achieved previously and the randomness in the updating step. Such a balance is part of the parameter setting issue. Each algorithm, in fact, requires the definition of several parameters roughly anywhere from 5 to more than 25 [124] and the values adopted can affect significantly the performance and no general rules exist to set them. It often happens that the iterative solution quickly converges in the neighbourhood of the optimum but then oscillates without converging to it. Search capability and convergence can be balanced or improved by modifying the basic formulation of the algorithms. In this regard, metaheuristics, being a set of numerical processes acting in series or in parallel, can be easily hybridised; namely, poor performing components can be substituted by better ones or even different algorithms can be combined in a sequential or parallel way, exploiting the best characteristics of each of them in addressing a specific task of the general problem.

As described in section 2.1.1 and 2.1.3, two typical SHM optimisation problems arise for network design and model updating. In network design any solution must satisfy constraints given by the demands, the requirements and the operational conditions of the sensing system. Thus, it is necessary to implement a strategy to check and reject the infeasible solutions. In literature, different criteria are commonly used to assess the suitability of the placement. Each of them stresses a specific aspect, thus the result could be affected by the objective function adopted [125]. Most of the methods developed so far addressed the modal identification of the monitored system, therefore, specific criteria aim at the identification of the DOFs which are excited more in the target modes through a proper norm of the Fisher Information Matrix (FIX), the Modal Assurance Criterion (MAC), the Singular Value Decomposition Ratio (SVDR) of the mode

shape matrix or the measured kinetic or strain energy per mode. Other criteria were developed to avoid redundant information [125], [126] or specifically for wireless sensor network requirements [127]. In this regard, several nature-inspired algorithms, like genetic algorithms, particle swarm optimisation, simulating annealing, monkey algorithm, ant colony optimisation, firefly algorithm, artificial bee colony, harmony search algorithm and wolf algorithm have been successfully applied [125], [128]–[131]. For this type of problems, metaheuristics outperform traditional methods despite a greater computational cost. Applications to large structures, like bridges [128], dams [132], high-rise buildings [129], prove their quality. Despite most of the methods developed analyse the best placement of single axis and single type of sensors, proper adjustments allow the OSP for multi-axial [132], [133] and multitype sensors [134].

In model updating, many nature-inspired optimisation techniques have been used. A review of the most popular ones can be found in [135]. As for the DI, which can be addressed through model updating approaches, Genetic Algorithms (GAs) and Particle Swarm Optimisation (PSO) are among the most known metaheuristics, with a wide literature and several applications, but many other similar algorithms have been applied for this purpose [136]–[139]. A complete review of this topic is missing, thus a proper comparison between different methods should be carried out. Regardless of the method adopted, the criteria to define the characteristics of the optimisation procedure are comparable. The inverse problem for DI can be formulated as the minimisation of an objective function, which is defined in terms of the discrepancies between features extracted by operational modal analysis and those computed using the numerical or analytical model. Nature-inspired metaheuristics can be used to calibrate the numerical model, analysing in parallel more possible solutions, belonging to the so-called population. Each of them is a vector representing the extent of the damage. This is often formulated in terms of stiffness reduction of the element: commonly the modulus of elasticity is used, but it is possible to consider a reduction in terms of area or momentum of inertia. The reduction is analysed through β_i , which is the ratio between reduced stiffness and undamaged stiffness, or α_i which is the ratio between stiffness reduction and undamaged stiffness, as follows:

$$[K_d] = \beta_i [K] = (1 - \alpha_i) [K] \quad (\text{Eq. 2.13})$$

When $\beta_i = 0$ the element collapses, vice versa when $\alpha_i = 0$ the element is undamaged. Both the factors vary in the range [0,1]. Several objective functions can be formulated based on the dynamical properties [135], [140]–[143]. The number of the decision variables is equal to the number of the candidate damage locations. Applications showed promising results, even though most of them resort to models with small number of elements, synthetic target data or small-scale laboratory tests, leaving some

doubts regarding the scalability. The performance of the algorithms, in fact, deteriorates as the number of variables increases. A common approach to overcome such a limit is to split the process into at least two stages. The first step aims to reduce the candidate locations. This can be done either through a coarse mesh, which is refined only across a candidate neighbourhood in the second step [144] or using the same discretization in both the stages but fixing the mechanical properties of the elements identified in the first step as undamaged. The metaheuristic algorithm is used in the second stage, after reducing the number of decision variables according to proper criteria [145]–[147], or in both [148]. An eventual third step can be added to perform a local search from the best solution of the global optimisation method [136]. The definition of the optimal population size is a challenge. The number of parallel solutions affects the result since a limited number increases the velocity of each iteration but reduces the rate of convergence and leads to less variety with the risk of a solution far from the optimum. On the contrary, a wide number turns the algorithm into a heavy and slow process [149]. Usually, a number higher than the variables (e.g. candidate damage locations) even up to 3 times is recommended. The evaluation of the objective function requires to run a FE analysis for each candidate solution at each iteration of the process. This is the main reason of the high computational cost. Thus, reducing the population size is essential to reduce the time consuming. Micro Genetic Algorithms (μ GA) aim at this using a limited population. The premature convergence is avoided by promoting the diversity through the initialization of a new population after a given number of iterations [150]. The initialization rule determines how the initial population of solutions is generated. In the canonical formulation, the diversification is guaranteed through a random generation, but deterministic rules and heuristic strategies can be implemented when more information on the problem is available. In DI, it is reasonable to assume that most of the elements are in an undamaged condition and that damage is located only in one or a few points. Therefore, it is possible to initialize the solutions to orient the search for low extent damages and a reduced number of damaged elements [151]. Although heuristic initialization is not so common in DI, some promising examples exist [152]. Another solution consists of introducing a penalty function to limit the number of damaged elements [153], [154]. Robustness is further increased through multi-objective optimisation by reducing the sensibility to the sources of error [144], [155].

During the development of the thesis plan, the author has explored the capability of metaheuristics for higher level DI. The main results have been published in [50], [156]. Although the present thesis focuses mainly on the development of a machine learning strategy for damage detection, it is worth mentioning the work done in optimisation as it is cast within the same global holistic SHM framework described in section 2.1. Damage detection is, indeed, a preliminary task in the post-processing stage and whenever

an anomalous behaviour emerges, the optimisation algorithm is run to identify the most likely damage scenario.

Nature-inspired SHM and classification

As seen in section 2.1.3, the data-driven approaches to DI rely on machine learning techniques, especially classification algorithms. Among them, neural networks [157], [158] are bio-inspired methodologies which have been largely exploited for damage detection, localisation and quantification. In the 1940s a pioneer work by McCulloch and Pitts [159] introduced the first mathematical model of a single neuron called, in the artificial system, processing unit or just unit. The neuron is an all-or-none, which fires when the weighted sum of the inputs from other neurons exceeds a threshold called bias. Different non-linear activation functions were progressively introduced following this first work, which used a linear function – commonly the sigmoidal activation function [160]. Most of the networks use additional layers hidden between the input and the output layers, and each of these layers receive the output of the previous one as input. In the field of SHM, multilayer perceptron (MLPNNs) with sigmoid activation function, trained by backpropagation, is the most common neural network architecture [157]. Among the possible network architectures, interesting applications adopted cerebellar model articulation controller network [161], counter propagation neural network [162], probabilistic neural network (PNN) [163], [164], radial basis function network (RBFNN) [165], recurrent neural network [166]. Further descriptions of these methods can be found in [103], [160], [167]–[169]. Despite different solutions were developed to achieve the same main purpose, there is a lack of studies comparing the performance of two or more different algorithms in SHM [157]. Rytter and Kirkegaard [165] provide a comparison between an MLPNN trained with backpropagation and a RBFNN, proving that the former is more reliable and the latter is strongly dependent on the training samples. Barai and Pandey [170] compare an MLPNN with a time-delay neural network, finding that the latter has a better performance though requires more time in training. Li [171] compared PNN and learning vector quantization neural network finding that the former performs better in damage localisation. PNN, usually, requires less time for training and performs better than the classic MLPNN [172]. State-of-the-art studies [157], [158], [172]–[174] provide a review of over 60 papers on DI through ANN from 1991 to 2015. As in the case of optimisation, the effectiveness of machine learning algorithms is affected by the parameter setting. In the case of ANN the most important are the number of nodes [175] and the training examples required. Moreover, these methods usually suffer data-over-fitting issues [172]. To reduce the size of the models and the computational effort, increasing the reliability even facing multiple local damages, an approach based on sub-structuring has been exploited [176], [177]. This resorts to a multi-stage strategy and to the training of different networks for different sub-

components of the global structure. Neural Networks have been also exploited for other SHM tasks. For instance, Bao et al. [58] used an autoassociative neural network (AANN) for noise reduction. The approach has an interesting bio-inspired paradigm since the detection imitates human biological vision and logical thinking to distinguish the sensor records patterns. AANNs have been also used for data normalization [178], [179].

Negative Selection Algorithms (NSAs) constitute another effective family of algorithms which are suitable for machine learning in SHM. Few applications exist in the literature and, given the promising results, more research is warranted. Since the goal of the present study is to further develop and test this methodology, a literature overview of NSAs is provided in the next sections.

2.2.3. Negative selection algorithms (NSA)

NSAs are a family of algorithms based on a minimal common framework, initially developed by Forrest et al. [180] and later improved with additional contributions [181], [182]. The complete framework of the process underlying the NSAs is composed of several steps that are collected in three main consecutive stages:

- (1) Representation;
- (2) Censoring;
- (3) Monitoring.

The first step to undertake is an overhead operation consisting in the definition of the feature space and the coding of the data set. During the so-called censoring stage, the NSA analyses the training data set in the feature space to generate the detectors. The detector set is the tool used for anomaly detection, being a set of elements which cover and identify the nonself portion of the space. The censoring should be carried out in a wider parameter setting strategy, in order to validate and optimise the detector set. Finally, in the monitoring stage, which is the actual classification stage, the new items extracted from the system under analysis are matched against the trained detectors, which bind to the ones that are likely to belong to an anomalous behaviour of the system.

Depending on the goal of the analysis, all the information collected in the previous steps can be post-processed for decision making. For instance, if the main goal is to compare different algorithm instances, in the last step their performances are ranked to infer their behaviour. Whereas, if the algorithm is already optimised and working, based on the results it is possible to decide whether to trigger an early warning or demand further numerical analyses or visual inspections. The latter option falls outside of the scope of the present study and it depends on the way the algorithm is integrated into a wider monitoring system.

The artificial negative selection process shows some specific features [183] that turn it very suitable for damage detection in the Civil Engineering field:

- The method works for one-class classification;
- The detection mechanism is based on a detector set;
- The information is represented in a negative fashion.

Being one-class classification methods, NSAs can be trained using samples from only one of the class, therefore they are suitable for application to real-world engineering cases where the only data available *a-priori* belong to the reference/healthy state. The detector set is suitable for distribution, mimicking the decentralised and distributable nature of the BIS: different detector sets can be used to classify the state of different components of the system or analyse different combination of features from the same system. Each set is autonomous and this distributed approach, where more sets are applied to the same system, reduces the classification failure and contributes to the effectiveness in real-world applications, especially for large complex systems [183]. Furthermore, representing the negative space, the detectors offer some chances to be generalised also to multi-class classification.

The computational usefulness and applicability of NSAs have been largely discussed and a unanimous result has not been reached yet. For a thorough analysis of the NSA theoretical background the reader can refer to [120], [183]–[186]. Although such concerns are relevant mainly from a computer science point of view, the consequences can influence the algorithm practical application. Hitherto, the discussion mainly involved the comparison of two processes, namely the negative and positive selection. Positive Selection Algorithms (PSAs) as opposed to NSAs are one-class classification method trained with self samples to provide an approximation of the self subspace instead of the nonself subspace. The two approaches share many features, being both straightforward instance-based learning techniques which carry out the classification based on geometric distance [185] and, in short, both aim at the definition of the boundary between self and nonself subspaces [120]. The analysis of their effectiveness led to an academic disputation [122], [181], [184]–[186]. An important aspect underlined by Stibor and Timmis [185] is that in literature more effort has been made in comparing different NSAs among each other, than in comparing them against completely different algorithms. In this regard, the first interesting comparisons of real-valued NSAs and PSAs with other methods are in [185], [187]. Applicability and effectiveness of NSAs has been demonstrated by [120], [183], whereas Stibor and Timmis [185] demonstrated the effectiveness of the PSAs and the curse of dimensionality affecting the NSAs. PSAs do not suffer the curse of dimensionality, but the complexity turns them unfeasible with a large number of training samples. In this case, according to Ebner et al. [122], the NSAs are more convenient; however,

although NSAs complexity also depends on the training set size, the problem is easily overcome by sampling or clustering the training set. Nonetheless, such a size reduction might filter out damage information content, reducing both false alarm rate and detection rate [185].

As suggested by Ji and Dasgupta [183], AIS is still a young field of research and NSAs deserve a more careful examination to answer such open issues. However, it is undeniable that the NSAs have been applied successfully to anomaly detection problems with several affinities to damage detection. It is also worth noting that, for the characteristics of the damage detection problems, the problem dimension is very rarely large enough to affect the NSA performance; furthermore, the definition of the detector set can effectively allow the localisation of the anomalies in the feature space providing some interesting results for an attempt of higher order classification. Thus, the NSAs method is considered a valid approach for DI.

Within the whole family of the NSAs, it is possible to define several algorithm versions and, within each of them, it is possible to define specific algorithm instances. This depends on two main parameters categories [89]:

- Finite domain parameters with no sensible distance metric or ordering;
- Real-valued domain parameters.

Operators and strategies belong to the first class. They are called qualitative, symbolic, categorical or nominal parameters and define the algorithm at the higher level: two algorithms are considered the same if their symbolic parameters are the same. The quantitative parameters belong to the latter class. They are also called numeric or ordinal and define a specific algorithm instance at the lower level. The differences in the setting of the numerical parameters can be due to differences in the problems analysed through the same algorithm.

The higher level components of NSAs are [183]:

- (1) Original problem space;
- (2) Data space representation;
- (3) Data coding;
- (4) Matching rule;
- (5) Detector generation/censoring.

These components are ordered in a top-to-bottom way as each one limits the setting of the following (e.g. the possible matching rules are different in distinct space representations). In Figure 2.8, the top-to-bottom organisation is clearly represented together with the main topics and methodologies mentioned in the following sections. The hybridization, namely the combination of a basic NSA with different

computational strategies, can be considered a further high-level algorithm strategy. This practice is extremely common to improve soft computing techniques and it is encouraged by the algorithms structure which can be decomposed into easy-to-share and adaptive sub-processes. In the present thesis, however, no hybrid strategy is directly adopted. A presentation of recently developed methods based on hybridization is provided by Ramdane and Chikhi [188].

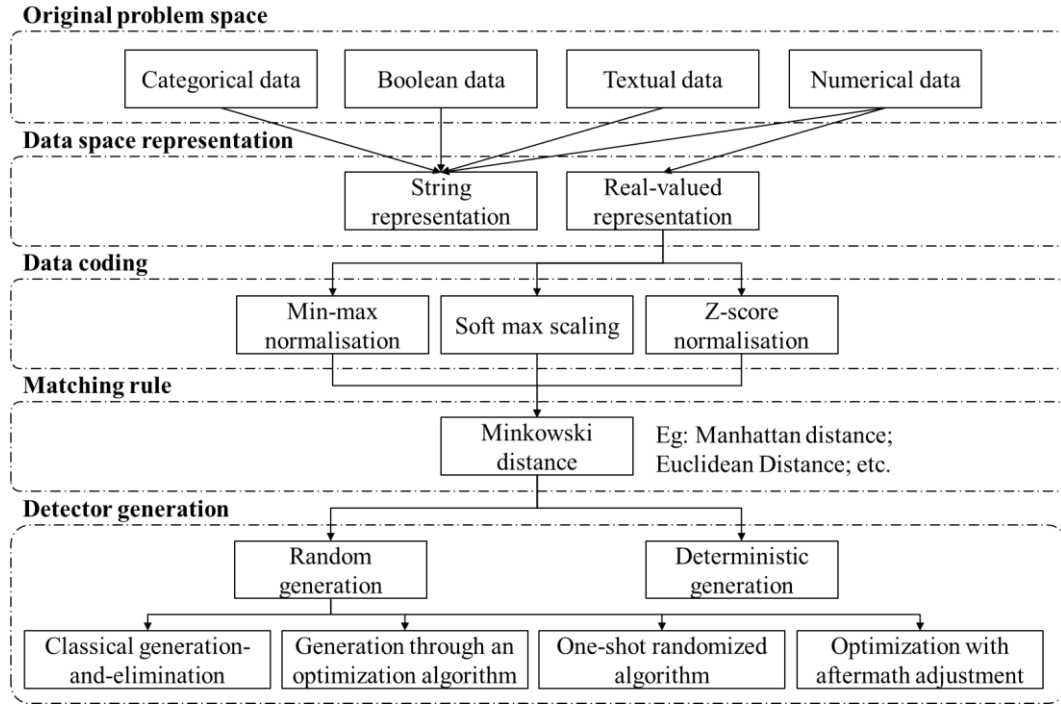


Figure 2.8: Summary of the topics analysed in the following sections according to the top-to-bottom order of the NSA components.

Original problem space, data space representation and coding

In the original problem space, the data belong to one out of four main categories, namely numeric, categorical, Boolean or textual [183]; moreover, two types of representation schemes for the data exist namely through string U_L^Σ or real-valued [183]. In the former, each element of the problem space is represented as a string U of length L over a finite alphabet Σ – e.g. in binary representation data are coded into strings over an alphabet given by $\Sigma = \{0,1\}$. All the four categories of data can be coded using a string representation; however, this is particularly suitable for categorical, Boolean and textual data. In the second representation scheme, the elements of the problem space are encoded as vectors of real numbers. If needed, a hybrid representation can also be implemented using items composed of different features belonging to both the set of real numbers and finite alphabets. Binary representation provided a strong mathematical background to analyse NSAs behaviour and to optimise their

performance. However, a comprehensibility problem emerges when numerical data are processed in low level representations. The encoding process, in fact, does not reflect the distance among the data in the original space as it introduces a different concept of distance and similarity, thus it is not possible to extract meaningful information from the result of the classification at the lower level [183]. Moreover, the string representation suffers of scalability issues, making it not feasible for large amount of data from complex systems [120]. The real-valued representation does not suffer from the previous issues, thereby being more suitable for numeric data. Considering that, in damage detection, the problem space is numerical and composed of all the possible real values assumed by one or more features of the system, a real-valued representation is adopted in the present thesis. Nonetheless, the input values domains are not directly used as a feature space. They are rather coded, by rescaling the monitored features values into a specific range. In the present study, the range adopted is $[0,1]$, therefore, the classification is performed into a unitary space, $U = [0,1]^n$, where n is the number of the features investigated. Experience demonstrated that this coding, also called normalization, increases both the speed of learning and the effectiveness of the algorithm. Several normalization algorithms exist and their choice can affect the classification performance [189]. For the purpose of this work three methodologies have been tested: mix-max normalization, soft max scaling and Z-score.

Min-max normalisation

Min-max normalisation (MMN) maps each value $x_{mi}, i = 1, \dots, L$, of the feature m into a new interval which ranges between 0 and 1 according to the following:

$$\bar{x}_{mi} = \frac{x_{mi} - \min_{i \in [1,L]}(x_{mi})}{\max_{i \in [1,L]}(x_{mi}) - \min_{i \in [1,L]}(x_{mi})} \quad (\text{Eq. 2.14})$$

The two bounds *min* and *max* are constant and based on the available training data. Though, during the monitoring, new elements, whose values exceed the predefined bounds, might appear. This “out of range” problem can be faced, first of all, by setting a larger range. Some authors suggest a 20% increasing, although it is evident that problem-specific knowledge helps defining a more suitable limit. However, it is still possible that “out of range” elements are collected. Thus, specific rules must be set to classify such values, according to the feature whose bound is exceeded. The rules are: (1) suspension of the anomaly evaluation, if it is not possible to infer anything regarding the behaviour of the system outside the known range of the feature; (2) “unhealthy” labelling of the element, if it is reasonable that a feature outside the range belongs to the abnormal state. For instance, if the feature is the n -th natural frequency, one can assume that a value outside of a 20%-extended range is likely due to damage onset. On the other side, measurements referring to a temperature value outside the known range cannot be classified with

sufficient confidence, as one cannot distinguish whether the anomalous behaviour is due to damage or to the temperature variation.

Soft max scaling (SMS)

Soft Max Scaling (SMS) normalization is a two-stage process. In the first stage, the value x_{mi} of the attribute m is mapped onto a range based on the distance between the mean value $\bar{x}_m$ and the standard deviation σ_m . This process exaggerates the distance of the values located far from the mean, resulting likely an index of anomaly:

$$\check{x}_{mi} = \frac{x_{mi} - \bar{x}_m}{\lambda \left(\frac{\sigma_m}{2\pi} \right)}, \quad \lambda = 3 \quad (\text{Eq. 2.15})$$

In the second stage, the new values are mapped onto the interval $[0, 1]$ through the logistic function:

$$\bar{\bar{x}}_{mi} = \frac{1}{1 + \exp(-\check{x}_{mi})} \quad (\text{Eq. 2.16})$$

Z-score normalisation

The z-score (ZS) is a two-stage process, similar to the SMS method. The first stage defines the range according to the mean value $\bar{x}_m$ and the standard deviation σ_m in a slightly different way:

$$\check{x}_{mi} = \frac{x_{mi} - \bar{x}_m}{\lambda \sigma_m}, \quad \lambda = 1 \quad (\text{Eq. 2.17})$$

Since the ZS does not have a specific output range, a second stage inspired by the SMS second stage is included to make this method suitable for NSA.

$$\bar{\bar{x}}_{mi} = \frac{1}{1 + \exp(-\check{x}_{mi})} \quad (\text{Eq. 2.18})$$

ZS is useful when the bounds of the distribution are not known, and the outliers dominate the data. SMS and ZS do not suffer from the “out of range” problem.

Matching rule, detector generation and censoring

The definition of the matching rule is essential to achieve a successful classification. This rule is composed of a distance measure and a threshold value. The matching rule allows for a partial matching, namely the matching happens when the monitored element is close enough to the detector: it is not necessary that the two are coincident. Requiring a complete matching, demands one detector for any possible nonself value, which for real-valued encoding, even considering the floating-point representation, is completely unmanageable.

The matching rule derives from the notion of shape-space, namely a combination of metric space and distance function, firstly introduced by Perelson and Oster [190] to describe the affinity between antibodies and antigens. The cross-reactivity or imperfect detection of the algorithm in real-valued representation is modelled by a detection threshold. In this way the detector is not only a point in the

feature space but a range of values with a given shape. This approach is called similarity recognition, opposed to the complementary recognition, closer to the real biological process as investigated by Hart and Ross [191], but harder to implement into an algorithm [192]. The choice of the matching rule is problem-specific and representation-dependent [183]. It is necessary to define a matching rule for both the censoring and the monitoring stage. The two rules can coincide or differ in the distance measure, the threshold or both. Furthermore, in the censoring stage, some NSA requires the definition of a matching rule for both the detector-self distance and the detector-detector distance. A general formulation of the distance D between two points in the m -dimensional space, $\mathbf{x} = \{x_i\}$ and $\mathbf{y} = \{y_i\}, i = 1, \dots, m$, is the Minkowski distance:

$$D(\mathbf{x}, \mathbf{y}) := \left(\sum_{i=1}^m |x_i - y_i|^\lambda \right)^{1/\lambda} \quad (\text{Eq. 2.19})$$

Different values of λ correspond to different distances or norms. The distance measure determines the shape of the area that each detector covers. Introducing the threshold r and considering a 2-D feature space, the Manhattan distance (or 1-norm, $\lambda = 1$) is a 45°-turned square of side $\sqrt{2}r$, the Euclidean Distance (or 2-norm) is a circle of radius r , the ∞ -norm is a square of side $2r$ and any norm between 2 and ∞ has an intermediate shape between the radius r circle and the edge $2r$ square (Figure 2.9) [193].

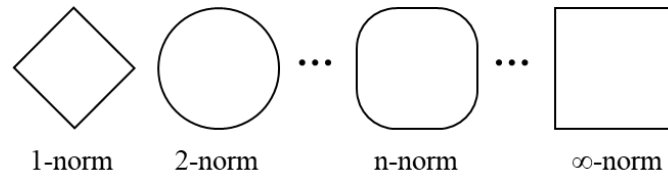


Figure 2.9: Detector shape in a 2-D feature space, adapted from [193].

The Euclidean distance is probably the most used: detectors based on such a rule in higher dimension spaces are hyper-sphere. For instance, in Figure 2.10, the generation and censoring stage is represented in a 2 dimensional feature space. In the example, the matching rule is based on the Euclidean distance and a threshold equal to the detector radius (r_{det}).

The detectors can be generated randomly or deterministically. In [183] four random approaches are identified: classical generation-and-elimination strategy; generation through an optimisation algorithm; one-shot randomized algorithm; optimisation with aftermath adjustment.

The deterministic generation proved to be a powerful strategy in lower representation where it is possible to define rules for the best detector placement. The strong theoretical background provided by the use of a finite alphabet, indeed, suggested that a deterministic generation allows to control the algorithm

complexity and the detector coverage [194]–[196]. This was done by relying on the concept of templates, namely building blocks for the strings, introduced by Helman and Forrest [197]. The first binary NSA versions with deterministic generation have been developed by D’haeseleer et al. [195]. Following, improvements have been produced (e.g. [194], [196]). The optimal generation rule depends on the matching rule adopted [198]. Most of the early research in the field was based on the binary representation; however, generalisations for a higher alphabet have been later developed, taking advantage of the concept of schemata, already used for evolutionary algorithms (as the genetic algorithm) [199], [200]. As for the real-valued representation, Li et al. [201] were the first ones to explore the deterministic generation.

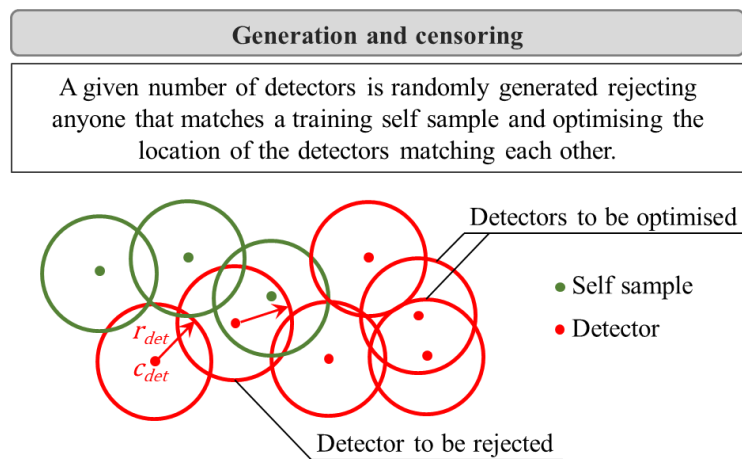


Figure 2.10: Representation of the generation and censoring stage in a 2-D space, considering a matching rule based on Euclidean distance and a threshold equal to the detector radius.

NSAs shortcomings

An extensive state of the art about the NSAs from a computer science point of view (e.g. [183], [188]) allowed to identify a series of shortcomings which must be addressed in the development of improved versions and considered in any application to real-world case studies. The main shortcomings are first summarised in Table 2.2 and then discussed in more detail; for the sake of simplicity, each shortcoming is associated with a code, used in the following thesis.

Table 2.2: Shortcomings of NSA with specific attention to damage detection.

Intrinsic issues	Problem-related issues	Detector-related issues	Monitoring strategy-related issues
S-I: Solution optimality	S-III: Data behaviour	S-VI: Nonself coverage	S-IX: Error rate
S-II: Curse of dimensionality	S-IV: Self approximation	S-VII: Detector overlapping	S-X: High-class classification
	S-V: Nonself samples	S-VIII: Detector set size	

The first two issues are intrinsic to the algorithm, but they are also common to many, if not all, soft computing techniques. More in detail, the solution provided by a real-valued NSA is sub-optimal (S-I) and the algorithm performance undergoes an unacceptable deterioration for larger problem dimensions (S-II). In the specific context of anomaly detection, the optimal solution can be seen as the correct boundary between the self and the nonself regions. Real-valued NSAs cannot guarantee the convergence to an optimal coverage with minimum overlapping among different detectors, irrespective of the parameter setting (e.g. matching rule, detector radius, optimisation of the detector locations, etc.) [100], [202]. Although the optimality cannot be demonstrated, many applications so far, in different fields, showed promising results and good performances, even in comparison with other largely diffused methods. The curse of dimensionality is determined by the significance of the data available for the learning process. Increasing the dimensionality of the problem, this data quickly becomes sparse. Stibor et al. [192] demonstrated that, for a given radius, the volume of the hyper-sphere (namely the detector) increases with the dimension until a peak then tends asymptotically to 0. Furthermore, increasing the dimension, the entire volume of the hyper-sphere tends to concentrate immediately below the surface. For small radii (e.g. $r \leq 0.1$), the peak corresponds to a negative value of the dimension, thus the detector volume always decreases by increasing the dimension. Such results explain the curse of dimensionality of the NSAs when high dimensional spaces must be covered using hyper-sphere detectors. In this case higher order norms, to measure the distance, demonstrated to be more robust.

As for the problem-related shortcomings, also these issues are not just NSA-specific and methodologies to tackle them can be developed for or shared with other numerical methods. The characteristics of the input data affect the NSA performance due to its lack of flexibility in the feature space over time (S-III). In other words, the learning process happens once offline, then the classifier defines the boundary of the nonself region and does not change during the online monitoring stage. Therefore, the detectors cannot deal with relevant changes either in the distribution of the self set with time (data dynamics) or in the nature of the self set in terms of analysed features. The former problem is related to the uncertainties of the investigated problem. The distribution of the analysed features, indeed, depends on factors that are not always known. In a real situation, the training data sampling might not be properly distributed over the domain of each factor, determining a biased definition of the self and nonself regions. During the monitoring stage, new data might be sampled from values of the unknown factors very different from the ones of the censoring stage, thereby determining a significant change of the distribution, without damage onset, and a systematic misclassification. Common NSA versions do not have a dynamic behaviour, namely the set of detectors cannot adapt easily to new collected information and a regeneration of the

detector set considering the new samples during the training might be necessary. A similar problem is due to changes in the type of features, namely if some feature is added to or removed from the feature vector. This can be easily explained through an example: assume that during a continuous monitoring of the first two natural frequencies of a structure as functions of the temperature, the third natural frequency is also registered but not at any time step. On the one hand, the detector set, trained in the 3D feature space of the 2 frequencies plus the temperature, cannot take into account the third frequency when its value is available, thus, the algorithm cannot directly take advantage of an improvement in the information content. On the other hand, the detector set, trained in the 4D feature space of the 3 frequencies and the temperature, cannot monitor any element whose value of the third frequency is missed; thus, the algorithm strongly suffers data anomalies.

In real-world applications, it is impossible to have access to a complete training set, namely coincident with the whole self set and presenting all the possible self patterns. Thus, it is necessary to define a good approximation of the self region based only on the samples available, which leads to the so called “boundary dilemma”. A sharp definition of the boundary based on a sample is, in fact, generalised to the whole population. The definition of the approximation quality and the inference of how well the classifier will perform on future samples that are not part of the training set is still an open issue (S-IV). Finally, a last problem-related shortcoming is due to the need of generalising one-class classifiers to binary classification problems. Better, it is not trivial to define a classifier which has an acceptable performance against nonself samples by training only on self samples. Thus, the lack of nonself samples for training or validation is a significant shortcoming in SHM.

Two sets of issues are specifically due to the NSAs framework, being related to the detector set and the monitoring strategy. Regarding the detector set, it is important to introduce the definition of holes or loopholes. These are nonself elements that no detector can recognize, even when the complete self set is available, and a proper matching rule is adopted. Mathematically, given the detector set D and the unitary space U divided into self space S and nonself space NS , the set that collects all the holes H is:

$$H \equiv \{x \in H \subseteq NS | H = U - \{S \cup D\}\} \quad (\text{Eq. 2.20})$$

In low level representation, it has been demonstrated that the holes cannot be avoided, however, they can be counted [183], [198], [203]. For real-valued representation, theoretical analysis is more complex and a common model taxonomy is missing, since most applications use their own version of NSA [183]. Random detector generations prevent a perfect coverage of the nonself region. Therefore, it is always required the numerical optimisation of the nonself coverage and the numerical minimisation of both the overlapping with the self samples and the overlapping among the detectors. This is a hard multi-objective

optimisation problem which has high runtime complexity [184], [204]. However, even for deterministic generation, in damage detection the training data are both incomplete and noisy leading to possible misclassifications close to the boundary region. In conclusion, it is impossible to deterministically define the number of detectors and the radius size to maximise the coverage (S-VI), nor it is possible to analyse, *a-priori*, the holes achieved with given detector characteristics [100], [202]. The presence of overlapping (S-VII) is usually unavoidable due to the shape of the detectors. For instance, given a two-dimensional unitary feature space, namely a square of side 1, circular or rounded detectors cannot cover the entire space without overlapping. Deterministic generation allows to predefine the characteristics of the overlapping area; whereas, for random generation, overlaps must be controlled and minimised. However, the overlap does not directly affect the performance in terms of classification. The reduction of the error rate, by achieving a better nonself coverage and by reducing the overlapping with the self samples, might influence the detector set size. This is in turn one of the factors governing the algorithm complexity (S-VIII). A significant complexity, namely a very large size of the detector set, not only makes the method worthless if compared to alternative approaches, but it also affects its scalability and its applicability. In this regard, a censoring strategy which leads to an excessive detector set is too time consuming if trained on a large sample set and, even if the detector set is properly generated, at the end it does not allow a nearly real time monitoring due to the time requirement to match any new sample against all the detectors.

For damage detection, it is necessary that the detectors provide an early warning, minimising the false negative errors and firing an alarm as soon as possible; however, the false alarms, namely false positive errors, generally have a relevant economic cost, due to the following inspections to verify the presence of the damage and the eventual closing of the structure (S-IX). Such an issue is related to the basic monitoring strategy which triggers the alarm whenever a new sample is classified as positive. As a rule, the problem can be framed into a more general reduction of the error rate that consists in finding a balance between false positives and false negatives, since it has been demonstrated that tuning the classifier to reduce the former leads to increasing the latter and vice versa.

Finally, it is worth noting that the basic monitoring strategy is suitable only for binary classification (S-X). This is not a shortcoming *per-se* since the problem addressed, namely the damage detection, is based on two possible conditions for the investigated system. However, the possibility to extend the methodology to higher order classification would make it more flexible and powerful in the cases where more information about the system behaviour allows the definition of more than two conditions.

2.2.4. NSA applications in SHM

As already mentioned in Section 2.2.3, the first Negative Selection Algorithm was developed by Forrest et al. [180] to address intrusion detection in computer systems, thus in a binary environment. The first application of the real-valued representation is provided by [100] with the so-called Randomized Real-valued NSA (RRNSA). The idea of generating real-valued detectors was informally proposed by Ebner et al. [122] without a numerical implementation. From then on, several versions have been developed to tackle the shortcomings that emerged in real-valued representation (see section 2.2.3). NSAs proved to be quite inexpensive and effective methodologies to address anomaly detection in several contexts. Possible applications are discussed by [188], [205]–[207]. Nonetheless, the applications to SHM for damage detection are still very limited. In Mechanical Engineering, NSA versions have been applied to vibration-based condition monitoring (e.g. [208], [209]). The results are promising in view of an application to Civil Engineering; however, this discipline assesses the state of machinery that, commonly, operates in constant and/or known operational and environmental conditions. NSAs were applied also to systems in variable operational and environmental conditions. The process of filtering out such effects with NSAs has been addressed by Surace and Worden [210], [211] over the data collected from the numerical model of a plane wing and of an offshore platform, as well as by Li and Ren [212] who used the numerical model of a beam. The features are, in the first case, the transmissibility function, simulating an experimental modal analysis with known input and, in the second case, the frequencies. In [213], the features are extracted by a Discrete Wavelet Transform and the method is applied to the IASC (International Association for Structural Control) – ASCE (American Society of Civil Engineers) SHM benchmark structure. In [214] the PZT acquisitions in several positions of the lamb wave generated by an input are used as features to detect and locate the damage in an aluminium plate.

Finally, it is worth mentioning the works carried out by other research groups that applied positive selection strategies to damage classification of different structural systems: Anaya et al. [215], [216]; Chen and Zang [217]–[221]; Mojtahedi et al. [222]; Zhou et al. [223], [224]. Although positive selection is a different process with respect to negative selection, the two methodologies share several common features, being almost one the opposite of the other: given the feature space and the training set, the negative selection aims at approximating the complement to the training set, whereas positive selection aims at approximating the training set itself.

In conclusion, very few applications of NSAs to damage detection exist, irrespective of the field (e.g. Mechanical, Aerospace or Civil engineering). Notwithstanding, the results are promising and warrant more research. In particular, the algorithms of this family are robust enough to provide good performance in

extremely different feature spaces. Nonetheless, an in-depth analysis of the algorithm strengths and weaknesses is missing to exploit the former and tackle the latter, and it is necessary to test and develop user friendly strategies to tailor the algorithm to the specificity of the case study through a well-designed parameter setting.

2.3. Conclusions

In this chapter the state of the art about Structural Health Monitoring (SHM) was presented, with special attention to Damage Identification (DI). The chapter described the demands, the requirements and the tasks that need to be fulfilled in the application of a damage identification strategy to any engineered system. Three main stages were pointed out, namely: (1) the design of a tailored monitoring system; (2) the pre-processing of the monitoring data (data acquisition, cleansing, fusion, reduction, normalisation and feature generation); (3) the post-processing interrogation of the features for the generation of a model able to identify the damage onset. Each stage was described extensively, driven by a holistic approach which intends to define a general SHM theory by focusing on the interconnected subtasks and stressing the open issues.

Within the field of biomimetics, new and well-established methodologies have been developed to address engineering problems by drawing inspiration from nature. Applications to SHM exist and were described in the present chapter, although the methodologies were usually developed for purposes different than SHM. Thus, a new framework to support a more tailored biomimicry to fulfil specific SHM needs was discussed. The present thesis has been elaborated within this wider and ambitious project. Addressing thoroughly all the SHM tasks would not be feasible, thus the goal of the present work is to focus on the exploitation of nature-inspired algorithms for selected SHM tasks. Among such tasks, trying to rate priorities according to a facility management perspective, the damage detection through classification algorithms has been chosen for a more in-depth analysis. Among the methodologies suitable to achieve this goal, Negative Selection Algorithms (NSAs) are adopted and tested in the present work. These algorithms belong to the paradigm of Artificial Immune System, whose natural source of inspiration was widely described in the final part of this chapter along with its artificial counterpart, followed by a review of the main published applications of NSAs to damage detection.

CHAPTER 3

Deterministically Generated Negative Selection Algorithm (DGNSA)

Abstract

The present chapter aims to describe the main contribution of the present thesis work, namely a complete damage detection strategy based on Negative Selection Algorithm (NSA). The features of the methodology are described with specific attention to the main issues identified in the state of the art. Moreover, a short guideline for the algorithm testing is provided at the end, in order to tackle essential concepts for the next two chapters, where the strategy is applied to different case studies.

3.1. Overview of the Damage Detection Strategy

In the context of structural damage detection, the final goal is the continuous online assessment of the system safety, by identifying at each subsequent data acquisition its condition. Offline assessment after periodic acquisitions can be framed as a particular case of the previous problem. The definition of the problem instance is essential and independent of the algorithm used for the damage detection; indeed, it is identified by the damage-sensitive features that are used for the classification. In the beginning of the process, a set of samples belonging to the reference (undamaged) state must be available as training set. However, the exclusive use of self data does not allow a proper validation of the algorithm performance. Therefore, the problem instance should be defined by producing also real or simulated feature samples in damaged states. As described in section 2.2.3, these aspects of the problem instance are associated with four main shortcomings (S-II, S-III, S-IV and S-V). The strategies to address such shortcomings are discussed in section 3.3. Once the feature space is defined and the samples collected, the damage detection strategy can be traced back to the execution of two following stages:

- (1) Offline detector censoring and parameter setting;
- (2) Online monitoring and damage detection.

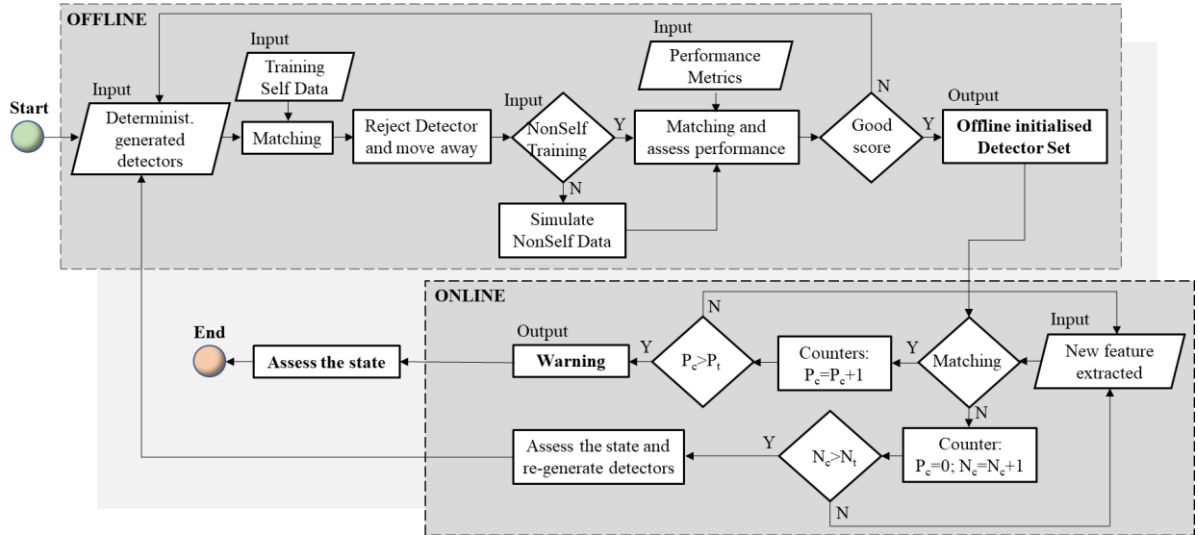


Figure 3.1: Framework of the algorithm, considering the offline parameter tuning and the online dynamic DGNSA.

Figure 3.1 shows the logical process flow underlying the application of the damage detection strategy. Moving from the shortcomings and limitations of existing procedures, a new version of the NSA has been developed to optimise the detector set, for low-dimensional feature spaces. Such a version is based on a deterministic generation strategy, further described in section 3.2.1, which is meant to tackle the poor nonself coverage (S-VI), limiting the overlapping (S-VII) and the detector size (S-VIII). According to this

methodology, a set of detectors is produced and matched against the self training samples. The detectors too close to the self training samples are rejected and the distribution of the others is optimised by a moving away process (section 3.2.1). As different parameter settings (in size and distribution of the detectors) are likely to provide different classification results, the initial generation is repeated according to a proper Design of Experiment (DOE) strategy (section 3.5) and the performance of the classifiers is analysed on a validating dataset where also nonself elements are needed. If real data are not available for the nonself condition, they must be artificially generated. The best classifier is identified according to one or more performance metrics in a deterministic way (section 3.5.2). The process is repeated until the metric score of the best classifier is considered to be good enough. Such a classifier will be the offline initialised detector set.

In the second stage, the classifier is used online for the continuous data monitoring. In the simplest case, this consists in matching any new sample against the detector set. However, a possible improvement allows to face the risk of false alarms (S-IX) and the case of dynamic problem domains (S-III), as discussed in section 3.2.2.

The last shortcoming that has been listed in section 2.2.3, namely the limit of NSA to the identification of two classes (S-X), is discussed in section 3.4. Assuming that the points belonging to different damage scenarios are clustered within the nonself space, higher order classification is viable by identifying in which region each specific damage scenario falls [225]. In the present work, two simple strategies to address this problem are discussed and tested.

After the description of all the features of the damage identification (DI) strategy, a short guideline for the algorithm numerical testing is presented in section 3.5. Although this topic might be trivial for many scholars, especially in Computer Science, it has been often underestimated in the Civil Engineering field. Thus, the information is provided to disseminate good practices and to support the comprehension of the following chapters where the applications are discussed.

3.2. Algorithm Instance

3.2.1. Deterministic generation

Deterministic generation in real valued representation has been first explored by Li et al. [201]. In the present work, a new strategy is developed. Given a matching rule based on Euclidean distance, the detector generation can be tackled as the problem of covering a square of unitary side length with circles, allowing overlapping to avoid holes. Few studies have already addressed the optimisation of the coverage of a 2-D unitary space with a given number of circles [226]–[228], but providing irregular distributions of

the centres. Hereafter, a regular grid is preferred because it is easier to implement and allows to use the same generation process for different number of detectors. Additionally, it is more suitable for nonself space structuring. The complete coverage of the feature space is still guaranteed, even without the optimisation, at the cost of a bigger overlapping area. The grid is generated by dividing each side of the unitary space in n intervals.

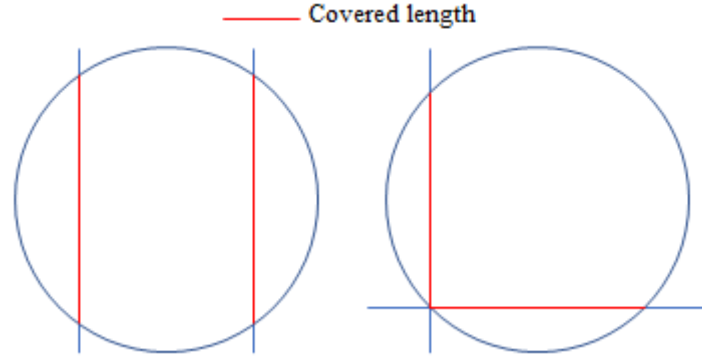


Figure 3.2: Maximum coverage of parallel and perpendicular lines with a circle.

In detail, the generation process consists of dividing the unitary flat surface into a given number of squares and assuming the detectors as their circumscribed circles. This approach is consistent with two well-known results of discrete geometry (Figure 3.2) [227]:

- A circle covers the maximum total length of two parallel lines when its position is symmetric;
- A circle covers the maximum total length of two perpendicular lines (x and y axes) when its position is symmetric and the intersection of the lines is on the boundary of the circle.

Given the number of divisions n , the side b of each inscribed square is:

$$b = 1/n \quad (\text{Eq. 3.21})$$

Thus, the detector radius r_{det} is:

$$r_{det} = \sqrt{b^2/2} \quad (\text{Eq. 3.22})$$

After the generation, the detectors are censored following a two-step procedure. First, they are matched against the training self samples and the ones that bind at least one self element are collected into a different temporary set. From this set, only the boundary elements are saved while the others are rejected. This means that, expressing the feature A of the 2-D space as a function of the feature B , the detectors with maximum and minimum value of A , for the same value of B , are preserved, whereas the others are discarded. Considering the distribution of the features, this approach allows to surround the samples. Then, the saved elements are iteratively moved away from the self data samples, as in the canonical form of the NSA, reducing the step size of the process at any i -th iteration by an adaptation rate η_i , to avoid

an excessive jump and to help the convergence to the minimum allowable distance. To this end, the following formulation is adopted [229]:

$$\eta_i = \eta_0 e^{-i/t} \quad (\text{Eq. 3.23})$$

where η_0 is the initial value of the adaptation rate and t is a parameter that controls the decay, named maturity age.

Once the moving away process is over, the saved elements are added back to the detector set. The pseudo-code of the censoring stage of the proposed deterministic NSA is shown in Table 3.1.

Table 3.1: Pseudo-code of the deterministic NSA censoring stage.

1.	Define (r_{self} , b , η_0 , t , <i>Censoring distance</i>) Generate the set D as a regular grid of detectors every b from $b/2$ to $(1-b/2)$ whose radius is $r = \sqrt{b^2/2}$ Initialize as 0 the counter <i>age</i> of the number of attempts of moving the detector
2.	Calculate, for each detector d the closest self element, <i>NearestSelf</i>
2.1.	If $distance(d, NearestSelf) < \textit{Censoring distance}$ Move d to a temporary set TD
3.	For each vertical column of the grid save the elements of TD with maximum and minimum normalized frequency and discard the others
4.	Calculate <i>NearestSelf</i> for each detector d in TD
4.1.	If $distance(d, NearestSelf) < \textit{Censoring distance}$ $age = age + 1$ $dir = \frac{d_i - NearestSelf}{distance(d, NearestSelf)}$
4.1.1.	If $age > t$ Move d to the set D
4.1.2.	Else $d_i = d_i + \eta_i \cdot dir$
4.2.	Else $age = 0$ Move d to the set D
5.	End

It is worth mentioning that this process is tailored to the distributions of the damage sensitive features which tend to be not separated. The main novelty of such a deterministically generated negative selection algorithm (DGNSA) consists in the use of a regular grid for the nodes which is generated following the same procedure for any predefined number of divisions. Furthermore, the simple coverage optimisation procedure, based on the moving away process, as developed in the first real-valued NSA [100], allows to avoid the presence of holes (S-VI) at least in the nonself region far from the boundary with the self space. The presence of holes in this boundary area is largely dependent on the training samples available and on the parameter setting. The method produces an acceptable amount of overlap (S-VII) among the detectors and, by parameter tuning, it can reduce the overlapping and the holes between detectors and self samples, without requiring complex optimisation strategies (Figure 3.3).

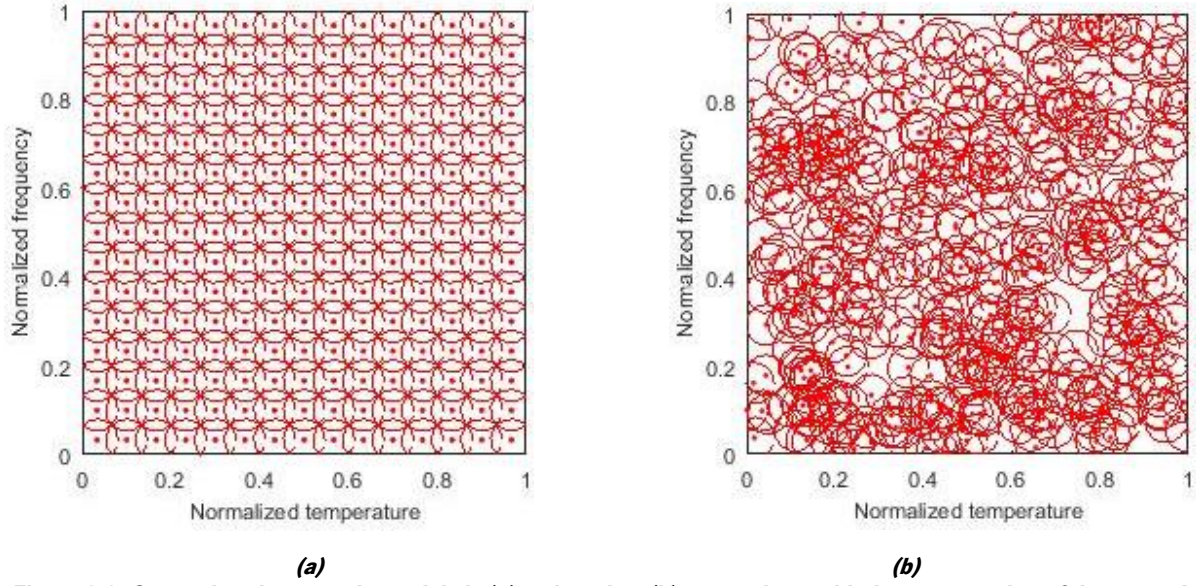


Figure 3.3: Comparison between deterministic (a) and random (b) generations, with the same number of detectors. In the random generation any distribution optimisation lacks.

Nonetheless, the deterministic generation increases significantly the computational burden, since the number of initialised detectors grows exponentially with the dimension of the feature space. Given a fixed number of divisions, n , for each dimension of the space, the number of detectors along that dimension is $d = n$ and the total number of detectors initialised in the m -dimensional feature space is d^m . Thus, the generation here adopted is strongly affected by curse of dimensionality (S-II) and is likely to produce an excessive detector set (S-VIII). However, based on the author experience and supported by many published studies in the field, it is assumed that two features are sufficient to carry out the damage detection. Limiting the problem size to 2-dimensional feature spaces is indeed a cost-effective approach for DI which prevents the negative influence of S-II and S-VIII. Moreover, the performance of the 2-dimensional detection can be improved by analysing in parallel more pairs of features.

The complexity of the NSAs is conditioned by the complexity of both the censoring and the monitoring stages. The former depends on the operations that are needed for detector generation and for detector training. During the generation, each detector is placed on the regular grid, thus, given m detectors, the complexity is $\mathcal{O}(m)$. During the training, each detector is compared with each training sample, thus, the complexity is $\mathcal{O}(m|R|)$, where R is the training set and $|R|$ its cardinality.

In the latter stage, any new monitored element $\mathbf{z}$ is matched against the detectors. The closest detector is identified and the element is classified as nonself if the following expression is verified:

$$\text{distance}(\mathbf{c}_{det}, \mathbf{z}) < r_{det} \quad (\text{Eq. 3.24})$$

Thus, for each new sample, the complexity of the monitoring stage is $\mathcal{O}(m)$.

3.2.2. Semi-dynamic DGNSA

A strategy to reduce the false alarm rate and to adapt the detector set with an increasing number of samples, during the monitoring stage, is here implemented by introducing two counters (one for the negative labels and one for the positive labels). Such a classification strategy has been applied similarly in literature to different fields of research and it is, here, modified and adapted to the damage detection strategy. The main inspiration source for the present application is the work by Hofmeyr and Forrest [230], with some differences that are further discussed at the end of this section. In the present work, such approach is called semi-dynamic, as the classifier is not completely static and a methodology is provided to define when a regeneration is required, nonetheless, the updating process cannot avoid an offline step once the detector set must be regenerated.

The counters must be updated after each matching with a new feature extracted. Whenever the new label is negative, the negative counter N_c is increased by 1, until it reaches a threshold N_t . After that, the system condition is assessed through alternative methods, to guarantee that the system is not in a false negative state. Similarly, whenever the new label is positive, the positive counter P_c is increased by 1, until it reaches a threshold P_t . After that, an alarm is triggered to warn the system analyst, namely the person in charge of interpreting the results of the monitoring for decision making. This counter allows to deal with outliers in the distribution of the self samples which could be mistaken for damaged features. Setting a very low positive counter threshold, eventually equal to 1, means that each false positive leads to a damage assessment, which could be expensive or even unfeasible. If a damage is really affecting the system, several consecutive measurements are likely to be classified as positive, increasing the reliability of the alarm. When the P_t limit is overcome, the manager carries out the assessment and, if the alarm turns out to be a false positive, the values are used as negative samples to carry out a new offline censoring stage. In this approach, it is convenient to set another threshold to reset the positive and negative counters. After the last positive (negative) label, if a number of consecutive labels of the opposite type, e.g. negative (positive), equal to the reset threshold are produced, the positive (negative) counter is set again to 0. The use of a counter is suitable for the specific problem under investigation, since a damage state is not naturally reversible, thus after the damage onset, a sequence of nonself points are expected [28]. As already mentioned, the present approach presents some differences from the ones found in literature, as in the work by Hofmeyr and Forrest [230], where the authors assign to each detector a positive counter P_c , thus the alarm is triggered only when the same detector matches P_t new samples. P_t is not constant, since it is reduced by one every time a detector matches a new sample, irrespective of the times that the same detector has been triggered before. This adaptive activation is

inspired by the biological concept of inflammation, since the multiple matching could be the sign of the damage outbreak even if the matching detector is not the same. In our case, the adaptive activation is achieved by resetting the P_c counter. A comparison with this or other similar strategies is not pursued in this research work and it is a possible future scope. The strategy based on the counters is exploited in the application to the case study described in section 5.3.

3.3.Problem Instance

3.3.1. Monitored Features

As discussed in the state of the art, vibration-based methods have been largely applied in SHM and DI, therefore, in the present thesis work, the damage sensitive features are extracted from vibration measurements, especially accelerations. In vibration-based DI, the analysed features, namely the Quantities of Interest (Qol) of the monitoring system, may belong to three possible domains:

- Frequency domain;
- Time domain;
- Time-frequency domain.

Qol belonging to frequency domain were the first investigated for fault detection. The analysis of the variation in modal parameters, in fact, dates back at least to the '70s of last century. Typical Qol were natural frequencies, modal assurance criterion (MAC), co-ordinates modal assurance criterion (COMAC), modal curvatures, modal flexibility, stiffness matrix derived from modal data, etc. [231]–[238].

The sensitivity of the features in frequency domain has been often doubted. Moreover, frequency and differences in modal coordinates can hardly provide information regarding the location, since different damage scenarios might cause the same amount of changes. The signatures, as the transmissibility functions, the power spectral densities or the simple Fourier transforms, can be analysed in specific spectral line ranges, after separating the whole signature into regions with an increasing damage-sensitivity, as the weak, fair or strong regions defined by Manson et al. [239]. Among the strong regions, one should prefer regions where damage and non-damage effects are uncorrelated, namely regions strong for the damage-induced changes, but weak for the non-damage induced changes. Defining these regions *a-priori* is often impossible, unless information about the effect of the damage on the signature is available. The main shortcomings of frequency-domain Qol is that, excluding artificial inputs, the ambient vibration and, consequently, the output are limited to the low-frequency range, which is less affected by localised damages. Furthermore, some loading can induce a (weak) nonlinear behaviour of the system (e.g. tall slender structures under wind loading, bridge under heavy traffic load, etc.), whereas many

algorithms assume a linear behaviour. The features based on Fourier analysis provide an average information over time, thus the time varying behaviour of non-stationary signal is ignored [240]. Finally, frequency-domain QoI might be strongly affected by sources of uncertainties rather than structural damage. Due to this, other QoI have been investigated over past decades.

The time-domain has been largely applied by using standardised vibration output signals. Time-dependent models are more suitable for the identification of time varying systems. The simple waveform comparison is not viable, since differences in the amplitude and the shape of the vibration signals due to sources of uncertainties are significant. The signal undergoes a time series analysis whose main objective is to make all the acquisitions comparable. Data samples of the same length are collected and time series models are applied (AR, ARX or ARMA). The QoI, as firstly developed by the group of Los Alamos, are the residuals or the coefficients of the model [241]. The time series modelling can be applied to the raw data or the raw data can be processed first (e.g. in [242] the pseudo-free response of the system is obtained by applying random decrement method to the vibration signal).

Recently, time-frequency domain methods have been also applied for system identification purposes. Short-time Fourier transform, wavelets and empirical mode decomposition are among the most common approaches in this domain. Such methodologies are particularly effective for non-stationary signal processing, overcoming the inherently global nature of Fourier analysis, which ignores the time varying behaviour [240]. An updated state of the art on such techniques and their related applications can be found in [240].

It is worth mentioning that the application of any NSA to a specific problem instance is independent of the features to be analysed. Following the approach here presented, the features are always scaled to the unitary space and processed mathematically without any attention to the physical investigated problem, that must be, thus, carefully considered by the analyst while defining the QoI and while interpreting the results. Indeed, the performance of the detection is strongly dependent on the features used.

Although the main scope of the work is to test and validate the effectiveness of the proposed algorithm for DI, a few considerations on the damage sensitivity of the possible adopted features are reported in the next chapters. The main shortcomings of the features in the frequency domain are well known, as described in the beginning of this section. Nonetheless, modal properties and dynamic signatures are still largely used for DI, hence, the decision of testing the algorithm on these QoI. In particular, natural frequencies are preferred, because they are easy to measure with a good reliability and a very reduced number of measurement points is enough for their estimation. Therefore, demonstrating a good detection performance based on frequencies, it is possible to design a sensor network where each sensor is

equipped and deployed in order to carry out an independent detection. To assure the reliability of such detector agents, only the lower frequencies are analysed. In fact, although higher modal frequencies are associated with local responses, thereby resulting more damage sensitive, their excitation can be difficult to achieve or even detrimental to the structure. However, despite their advantages, natural frequencies present also possible shortcomings, especially for the global modes in the low frequency range, since their shift due to damage might be negligible and they might suffer from a lack of spatial information. These shortcomings might be overcome using more frequencies, since different damage scenarios will cause different combinations of shifts [157], as they will induce non-uniform and damage-specific variations to the vibration modes. However, using more frequencies but limiting the feature space to two dimensions is a practical issue, unless considering more classifiers analysing different pairs in parallel.

In the present study, frequency distances (e.g. $f_2 - f_1$) are used to concentrate the information regarding 3 or 4 frequency shifts into a 2-D space. Another class of features that are tested in this research work are the vibration signatures, such as the Fast Fourier Transform and the Power Spectral Density of the accelerometer records. In this case, the 2-D space is given by the frequency lines and the amplitude. The main advantage is that vibration signatures not only can be analysed separately by each sensor, but, potentially, have a larger information content than natural frequencies.

These QoI are extracted from the structural response time histories, commonly in terms of accelerations. This is done through modal identification techniques. Several algorithms are suitable for this kind of process, belonging to two main families:

- Input-output identification;
- Output only identification.

The former family includes the first developed methodologies for modal identification. All of them need the acquisition of the time histories of both the input excitation and the output structural response in a set of meaningful points distributed across the system. The algorithms operate on the Frequency Response Functions (FRFs) or on the Impulse Response Functions (IRFs). Further criteria to classify the methods for input-output identification are based on the domain (frequency or time), the type of formulation (indirect or direct), the number of degrees of freedom (single or multiple), the type of estimates (local or global) and the number of inputs/outputs that can be managed (Single Input-Single Output – SISO, Single Input-multi Output – SIMO or Multi Input-Multi Output – MIMO). The most common methodologies are listed in Table 3.2, together with their main characteristics. Nowadays, a very wide literature on the topic is available, for more details, the reader can refer to [243], [244].

Table 3.2: Input-output identification methods (adapted from [245]).

Domain	Method	Formulation	DOF	Estimates	Input/Output
Frequency	Peak Picking	Indirect	SDOF	Local	SISO
	Circle-fit	Indirect	SDOF	Local	SISO
	Inverse	Indirect	SDOF	Local	SISO
	Dobson	Indirect	SDOF	Local	SISO/SIMO
	Nonlinear-LSFD	Indirect	MDOF	Local/Global	SIMO/MIMO
	Orthogonal polynomial	Indirect	MDOF	Local/Global	SIMO/MIMO
Tuned-Sinusoid	Mau	Indirect	SDOF	Local	SISO
	Asher's	Indirect	MDOF	Global	MIMO
Time	Complex Exponential	Indirect	SDOF	Local	SISO
	LSCE	Indirect	MDOF	Global	SIMO/MIMO
	Ibrahim	Indirect	MDOF	Global	MIMO
	ERA	Indirect	MDOF	Global	MIMO
	ARMA	Direct	NDOF	Global	MIMO

The output-only identification is particularly relevant in the Civil Engineering field, since the controlled artificial excitation of large systems is often economically and practically unfeasible. These methods rely on the exclusive acquisition of the output vibration response of the system to freely available natural excitations, random in time and in space (like traffic, wind, human walking and micro-tremors), assumed as stationary Gaussian white noise stochastic processes in the frequency range of interest for the structure. These methods are further classified depending on the domain in which they operate (frequency or time), see Table 3.3 for some examples [246].

Table 3.3: Output only identification methods (adapted from [245]).

Frequency-domain	Time-domain
Peak picking	Random Decrement
Frequency Domain Decomposition	Recursive Techniques (ARMA)
Enhanced Frequency Domain Decomposition	Maximum Likelihood methods
Polimax	Stochastic Subspace Identification Methods

The analysis of the performance of the feature extraction algorithms is out of the scope of the present work, as the focus will be on the analysis of the extracted features for damage identification through DGNSA. Nonetheless, it is worth mentioning that different strategies have different advantages and disadvantages and it is not possible to univocally identify a method that outperforms all the others on all the possible datasets [247]–[250]. Therefore, the effect of the uncertainties due to the adopted feature

extraction process and in general its influence on the damage identification performance, for the improvement of the global detection strategy, are recommended as future scope.

3.3.2. Self space approximation

The training set represents all the available information regarding the problem instance in the design stage of the classifier. Commonly, it is composed of self samples which constitutes an approximation of the whole self space, namely the space collecting all the possible values of the analysed features in the normal state. The whole self space has potentially infinite elements, whereas the training set is inevitably a partial and sparse approximation of it. Irrespective of the number of samples available, in fact, it is not possible to guarantee that any unknown normal state point will fall within the approximated space. Furthermore, in SHM, creating a training set means collecting as many records as possible in all the possible expected conditions during the system service life. Thus, the sampling of many records along the time is required, with relevant economical costs. Two extreme situations exist:

- A large number of samples is already available from previous monitoring campaigns;
- The monitoring campaign must be designed and no information about the system is available.

In the former case, an approximation of the self space through less samples can speed up the numerical processes, reducing the memory requirements for the data storage and the size of the problem, thus, the computational burden during the analysis. In the latter case, reducing the number of samples to the minimum reduces the costs due to the acquisition and the delay in the following tasks when the monitoring is part of a larger project. In many engineering applications there are, indeed, limitations on size and cost of the testing programmes.

During the development of improved strategies for the self space approximation through a small number of samples, two fundamental questions must be addressed:

- How adequate is the approximation provided by the training set?
- How many samples are needed to provide an adequate approximation?

In the feature space, this can be described as the geometrical problem of approximating an area by using only a limited number of points belonging to it. Two diametrically opposite approaches for the approximation of the self space are discussed hereafter. The first one is deterministic, whereas the second one is stochastic. The two essentially differ for the nature of the analysed variable. In the former, the goal is to use the points to create an approximate self space as the region within the envelope of the points, irrespective of their distribution statistics. In the latter, the self space is considered as a stochastic variable

whose probability density function is investigated based on the available samples. Hereafter, the two approaches are described, more details are provided in the application to the case study in section 5.2.

Deterministic approximation

The hypothesis underlying the deterministic approximation is that the self space corresponds to the area within the most shrunk boundary which envelopes the available points. Due to the well-known influence of the environmental and operational conditions on the system vibrational response, the acquisition cannot be carried out in a completely random way, but they must be representative of the expected variation range for such conditions, the goal being the identification of the extreme points of the space (e.g. maximum and minimum frequency bounds). The most unfavourable situation is when no previous information regarding the frequency bounds is available. Assuming that the system response is particularly affected by the temperature, the behaviour in the expected range of temperature must be investigated. Afterwards, this is subdivided in subranges and random acquisitions are sampled within each subrange. In the case studies that are presented in the next chapters, the effects of both the number of ranges and the number of samples within each range are analysed, filtering out the systematic error due to randomness by repeating the tests with different random samples. The random generation of the samples simulate a real-world modal identification with the only condition that it is carried out in a pre-defined interval of temperature. All these instances are compared with the exact self space, that in a real case study is unknown, but for the validation is here predefined. The metrics used for the comparison are the similarity between approximated and actual areas, their intersection and the amount of nonself space covered by the self approximation. The similarity is expressed as:

$$Similarity = 1 - \frac{Union - Intersection}{Union} \quad (Eq. 3.25)$$

The first two metrics are an estimate of the percentage of true negatives correctly classified, whereas the third is an estimate of the percentage of false negative errors. For each simulation, the area within the samples is calculated. This value is used to define the incremental area growth which is equal to the increment of area due to the addition of samples divided by the number of samples added.

Stochastic approximation

The stochastic approximation relies on the distribution of the self samples in a real case-study. Self samples, in fact, fall into a relatively stable range presenting dense areas of overlapping points surrounded by sparse areas of rare ones. The stochastic approximation aims at defining the probability distribution of the full self population, based on the distribution of a samples set. Although only few studies in the field of NSAs address this specific problem, some techniques have been here collected and tested. All of them define an approximation of the self set as the area within a circle of given radius around each known

point. Different samples have different radius: bigger in the dense region and smaller in the sparse region that is close to the boundary. In particular, three approaches selected from the literature are compared in the next chapters. The first, called affinity method [251], is based on the affinity density of each sample A , which is calculated as:

$$\rho(A) = \frac{1}{N-1} \sum_{i=1}^N \frac{1}{d(A, X_i)}, (X_i \neq A) \quad (\text{Eq. 3.26})$$

where $d(A, X_i)$ is the Euclidean distance between the sample A and all the other $(N - 1)$ samples. In this case, the sample radius is given by:

$$r(A) = R \frac{\rho(A) - \rho_{min}}{\rho_{max} - \rho_{min}} \quad (\text{Eq. 3.27})$$

where R is the radius value for the maximum affinity density sample, while ρ_{min} and ρ_{max} indicate the minimum and the maximum affinity densities for the sample, respectively.

The second is called Gaussian method [252], since the probability distribution of the samples is approximated by a Gaussian multivariate distribution. Once the mean of each feature and the covariance matrix are known, the joint gaussian probability density function pdf_A of sample A is calculated. The sample radius is:

$$r(A) = R \frac{pdf_A - pdf_{min}}{pdf_{max} - pdf_{min}} \quad (\text{Eq. 3.28})$$

where R is the radius value for the maximum probability density sample, whereas pdf_{min} and pdf_{max} are respectively the minimum and the maximum probability densities of the sample set.

The third and last approach applied for the stochastic approximation of the self space is the B-NN-ASR algorithm, developed by Zhang and Luo [253]. According to this method, for each self sample, the distance to the k nearest neighbours is calculated and the vectors, v_i , connecting the sample to these neighbours are defined. The sample is classified as boundary or internal according to the angle between such vectors and the vector sum, $v_{sum} = \sum_i^k v_i$:

$$ang_i = \frac{180^\circ}{\pi} \arccos \left(\frac{v_i \cdot v_{sum}}{\|v_i\| \times \|v_{sum}\|} \right) \quad (\text{Eq. 3.29})$$

The sample is internal if $\max(ang_i) \geq \alpha$. In [253] a value of α equal to 150° is recommended to distinguish between internal or boundary samples. Internal samples have a radius equal to the distance to the $k/2$ -th nearest neighbour, whereas a boundary sample A is moved in the direction of the closest internal nearest neighbour B to the point $\frac{A+B}{2}$ and its radius is:

$$r(A) = \frac{dis(A, B)}{2} \quad (\text{Eq. 3.30})$$

The performance of the different methods is evaluated in terms of self coverage (SC) and nonself coverage, (NSC). Namely, given a set of self and nonself elements, the coverage corresponds to the number of samples falling within the circles approximating the self set:

$$\begin{aligned} SC &= \frac{\text{N}^\circ \text{ of self samples covered by the approximate self set}}{\text{Total N}^\circ \text{ of self samples}} \\ NSC &= \frac{\text{N}^\circ \text{ of nonself samples covered by the approximate self set}}{\text{Total N}^\circ \text{ of nonself samples}} \end{aligned} \quad (\text{Eq. 3.31})$$

The best scenario for the performance is given by an $SC = 1$ and an $NSC = 0$, meaning that the approximated area includes all the self samples without covering the nonself region.

These two metrics are chosen since, in this case, the circles present a large overlapping area and a direct estimation of the covered self space is not as straightforward as in the deterministic approximation.

3.3.3. Nonself data required for assessment

The main issue of one-class classification is the complete lack of information regarding the values of the features when the system is damaged. Indeed, when a classifier is applied to the damage detection, any sample may belong either to the healthy state (negative feature) or to the damaged state (positive feature). The families of possible errors are two: false positive (FP), namely normal states labelled as abnormal, and false negative (FN), viz. abnormal states labelled as normal. Similarly, the correct classifications can be only two: true positive (TP), namely abnormal states labelled as abnormal, and true negative (TN), i.e. normal states labelled as normal. Experience demonstrated that, changing the algorithm parameters values (e.g. self radius, censoring distance, etc.) the number of FPs and FNs varies inversely. This is due to the complex approximation, through the detectors, of the border between self and nonself spaces. In more detail, a parameter setting which leads to a larger coverage of the border area increases the TPs (thus reducing the FNs), but is more aggressive, being more likely to identify negative samples close to the border as anomalous (e.g. more FP errors). On the opposite, a parameter setting which leads to a less coverage of the border has a more tolerant behaviour: many negative samples close to the border do not trigger its response (e.g. less FP errors), but also the samples belonging to a damage state closer to the self space are mistaken for self (e.g. more FN errors). Therefore, a parameter setting carried out simply based on self samples might minimise the number of false positive errors, but cannot minimise the number of false negative errors, actually it might rather increase it. To address this shortcoming, nonself samples can be artificially produced by resorting to a numerical model of the system. However,

this involves several problems as, for instance, the reliability of the model itself in the presence of many sources of uncertainties, or the representativeness of the simulated damage scenarios. Therefore, in the present work, a data-driven methodology to simulate nonself samples is tested as well. This consists in sampling a set of outliers based on the joint probability bivariate distribution for the two investigated features. The range of one feature is divided into intervals and in each interval the marginal distribution of the second feature is calculated. Based on that, the outliers are an ordered pair with:

- First value randomly sampled with uniform distribution in the first feature interval;
- Second value placed out of the following range:

$$\begin{aligned} sup &= Q_3 + 1.5(Q_3 - Q_1) \\ inf &= Q_1 - 1.5(Q_3 - Q_1) \end{aligned} \quad (\text{Eq. 3.32})$$

where Q_1 and Q_3 are the first and the third quartile of the marginal distribution of the second feature in any interval of the first feature.

3.4. Higher order DGNSA-based damage identification

Recalling the organisation of the DI process into five main goals, according to the Rytter's hierarchy [29], [30], the methodology described so far addresses the first identification level, namely the detection of the damage existence. This is a basic level that any complete damage identification strategy should provide in an online, real-time way, limiting the false alarms and the need for longer inspection and/or more complex analyses, only when the structure is very likely damaged, for the higher levels (e.g. localisation, quantification and prognosis). Developing a fast and reliable approach for the detection is still challenging, hence, it is the main focus of the present thesis. Nonetheless, the DGNSA presents some interesting features that make it suitable for higher order identification. This scope is considered secondary, namely, the main goal of the algorithm is to provide a fast binary response regarding the soundness of the system. However, in some situations, it can also provide information regarding the most likely class of damage among a set of previously defined damage scenarios. To that end, two strategies have been developed and tested to extend DGNSA application to higher identification levels, namely for damage localisation, classification and quantification, showing some promising results, which are reported in the next chapter.

Substructuring strategy

Experience demonstrated that unlike features can be more or less sensitive to different damage scenarios. This distinct sensitivity, which is clearly shown in the applications that are discussed in section 4.2, has a twofold importance:

- On the one hand, it allows to define a likelihood of the damage presence, given the detection in some of the substructures;
- On the other hand, it enables to extend the identification to higher levels.

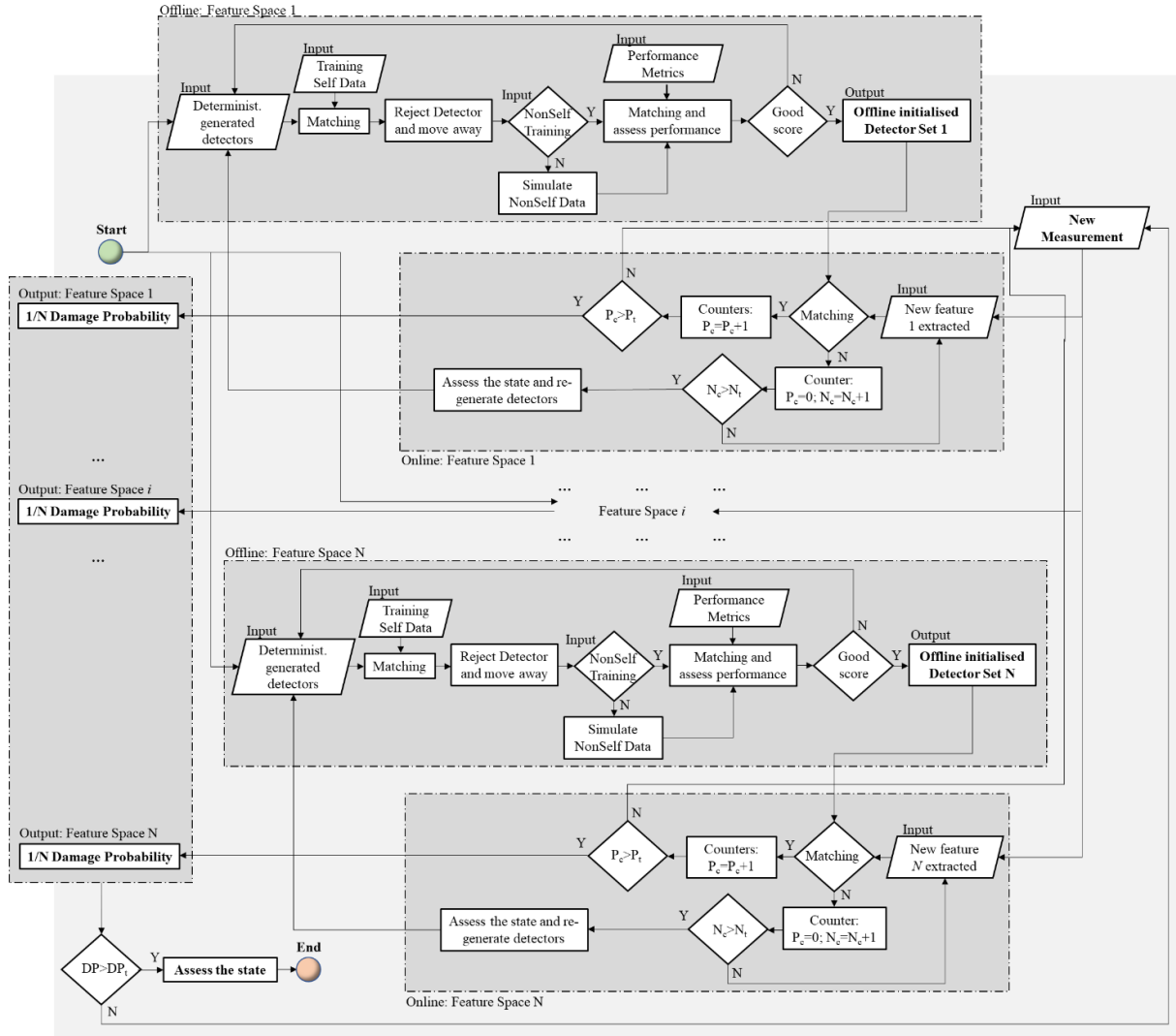


Figure 3.4: Framework for substructured NSA

As shown in Figure 3.4, the substructuring strategy consists of a monitoring stage with more classifiers in parallel (e.g. one classifier analysing the 1st-2nd frequencies feature space, one analysing the 1st-3rd frequencies space, one analysing the 2nd-3rd frequencies and so on). Each classifier is a substructure of the global classifier and works as in the basic process flow of Figure 3.1: it is optimised and tailored to a specific feature space and labels the system state based on each new acquisition. Since the classifiers analyse different feature spaces, the labels are likely to be different. On the one hand, the simple ratio between the number of positive labels and the total number of classifiers can be used as an estimate of the likelihood of the damage. On the other hand, since each feature space is more sensitive to some

classes of damage, knowing which features lead to a positive classification may allow to infer which damage scenarios are likely to be present. This involves a supervised learning process; thus, it requires that the response under different damage scenarios is known. Any inference regarding possible damage location and extent is limited by the classes of damage investigated.

The case-study application described in the next chapter will help to better understand the mechanism.

Signature analysis

The second strategy involves only a single classifier. When the features used for the classification are vibration signatures (e.g. the power spectral densities), each monitored status has more points in the feature space (e.g. a value of amplitude per each recorded frequency), thus it is possible to define, for both FPs and TPs, an “anomaly score” equal to the number of elements of the signature matched by at least one detector. The distribution of this score for a specific class of damage (e.g. a specific combination of damage location and extent) can be visualized through histograms of occurrences, namely the number of measurements with the same anomaly score, eventually reported as percentage. A schematic example is provided in Figure 3.5a. Here, 100 samples belonging to the same class are classified. For each of them, the anomaly score is evaluated. Out of 100 elements, it results that 24 have an anomaly score of 6, 28 elements score 7, 26 score 8 and 22 score 9, meaning that 24 elements have been matched 6 times by at least one detector, 28 elements have been matched 7 times by at least one detector, and so forth. The frequency-domain distribution of the elements triggering the detectors can be displayed through the matching chart, which is a 2-D plot of the monitoring data elements per each damage scenario against the frequency lines. The purpose of this chart is to allow the identification of the anomalous points of the spectrum, namely the points modified by the damage. In Figure 3.5b, the process to extract the matching chart from the signatures under classification is visually schematised. The points of the signatures in a portion of the peak match with at least one detector and, therefore, are classified as nonself. The anomalous frequency lines are reported in black in the matching chart.

Differences in the distribution produced by a classifier for the different damage scenarios have a twofold importance:

- On the one hand, it allows to define kind of quality index of the classifier itself and a qualitative estimation of the damage extent during its evolution;
- On the other hand, it allows to extend the identification to higher levels.

For instance, in Figure 3.6, the distribution of the anomaly score for a single classifier, considering 4 different classes of damage, is reported. The signature presents 32 points, that is, thus, equal to the

maximum anomaly score possible. The mode of the distribution for the damage class 1 is 3 points, for the class 2 is 14, for the class 3 is 21 and for the class 4 is 27.

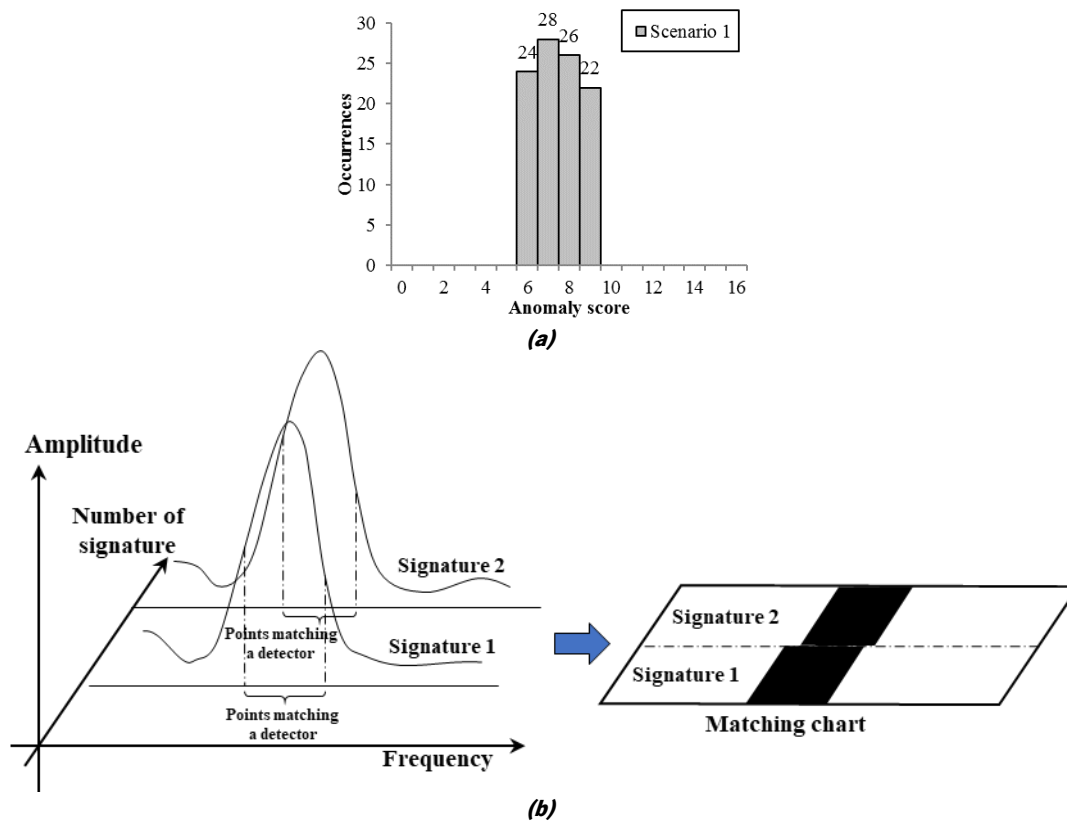


Figure 3.5: Examples of the graphs used to report the experiment outcome: (a) occurrence histogram; (b) extraction of the matching chart.

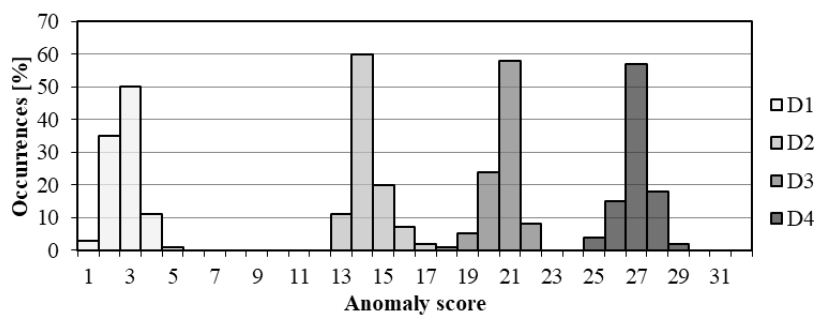


Figure 3.6: Distribution of the anomaly score of a classifier for 4 different classes of damage.

On the one hand, it is found, in the analysed case studies (see section 4.3), that a larger number of points of the signatures bound by at least one detector corresponds to a larger extent of the damage. Thus, the anomaly score distribution can be seen as a qualitative measure of the distance between the healthy and the unhealthy conditions. In the example, the damage class D4 is likely to be the most severe among the 4 scenarios. Nonetheless, at this stage, the measure is not absolute in quantitative terms, as an increasing damage severity induces a non-proportional increase in the anomaly score.

On the other side, based on the 4 distributions that were known during the training, it is possible to classify any new sample, not only as damaged or undamaged, but also to determine its likelihood to belong to a specific damage class. This is pursued by considering the score for each damage scenario as a discrete random variable, thus defining its probability mass function. The distributions are approximated to Gaussian ones. The strategy requires a supervised learning and, as in the previous strategy based on substructuring, any inference regarding possible damage location and extent is limited by the classes of damage whose samples were available during the training. When a new element is monitored, based on its anomaly score, it can be classified as belonging to one of the known scenarios with a given probability, by using the Z-score. Namely, given an element x , and a damage scenario i , its Z-score corresponds to the standardized statistic:

$$Z_i = \frac{x - \mu_i}{\sigma_i} \quad (\text{Eq. 3.33})$$

where μ_i and σ_i are, respectively, the mean and the standard deviation of the scenario.

The two-tailed p-value of a Z-test statistic is assumed as an approximation of the probability that the element belongs to the given scenario. Actually, the Z-test, given a sample set whose mean is μ_0 , checks the null-hypothesis that this value is equal to the mean of a given normal distribution, $H_0: \mu_0 = \mu_i$

Whenever Z is such that the p-value $P(Z < -Z_{\alpha/2} \text{ or } Z > Z_{\alpha/2}) = \alpha$, for a given significance level α , the null-hypothesis is rejected. However, in this case, a single value rather than a sample set is compared to the known distribution.

Finally, this approach allows also to compare the algorithm instance. Given two algorithm settings which provide the same number of correctly and incorrectly classified samples over the same data set, thus the same performance measure, it is assumed that the one with higher anomaly score for the same damage scenario and well-separated distributions for different scenarios is likely the best. It is important to consider together the value and the distribution. For instance, Figure 3.7 shows the results of two algorithm parameters' instances (named classifier 1 and 2) against the same training and validating data. Checking the histogram of the classifiers 2, one might mistake the higher anomaly score against the scenarios D1, D2 and D3, for a sign of good performance. Quite the opposite, the distributions of the anomaly score occurrences for the true and the false positives almost overlap completely, and the points of the signatures detected as anomalous for the different damages are almost the same, being impossible to distinguish whether a new sample belongs to a specific damage class rather than another.

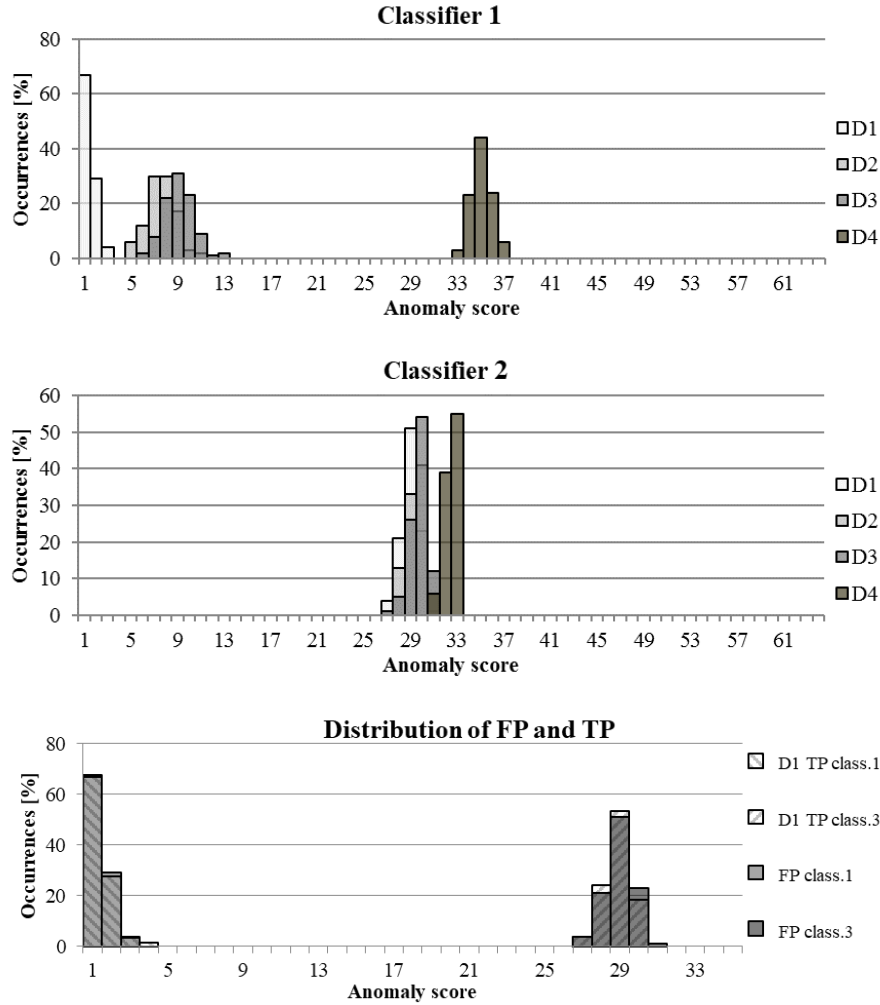


Figure 3.7: Distribution of the anomaly score of two classifiers for 4 different damage scenarios.

It is worth mentioning that the anomaly score can be calculated not only for the correctly classified nonself elements, or True Positives (TPs), but also for the incorrectly classified self elements, or False Positives (FPs). In the example reported in Figure 3.7, it emerges that there is a high risk of false alarm for both the classifier, as the undamaged state is easily mistaken for the scenario D1. For classifier 1, for instance, almost the 80% of the signatures belonging to the first damage scenario are labelled nonself through the matching of a single element of the signature. However, unlike classifiers 2, the distributions of the anomaly score occurrences, for the other damage scenarios, in classifier 1 are well separated, making the higher order classification easier and thus turning the classifier 1 more suitable for this purpose.

3.5. Algorithm testing

All the conclusions drawn in the present work are based on algorithm numerical experiments. An experiment is a set of tests run under controlled conditions for a specific goal [254]–[256].

Experimentation is composed of the following steps [255]:

- (1) Goals identification;
- (2) Performance metrics;
- (3) Experiment design and execution (including algorithm factor levels definition, data analysis and conclusion);
- (4) Results report.

Such tasks are briefly explained in the following sections with specific attentions to the applications that are presented in the next chapters.

3.5.1. Goals

The goal defines the specific question which motivates the experiments [257]. The algorithmic studies have three main aims [258]:

- Demonstrate the feasibility of a newly developed algorithm;
- Improve and modify the algorithm or its components;
- Compare different algorithms performance.

Several experiments are carried out in the present work, aiming at the three goals that are mentioned above. Indeed, in the first stage, a novel version of NSA is proposed and tested. Nonetheless, most of the analyses discussed in the next chapters are aimed at the second and third goals, namely the modification of the algorithm components and the comparison against different methodologies. Both are performance comparisons, thus the experiments are analysed in a similar statistical framework. However, in the first case, different configurations of the same DGNSA are compared to calibrate the inner parameters values; whereas, in the second case, instances belonging to different families of algorithms are compared to identify which one has the best performance and possibly why. The comparison must be fair and aim at a better understanding of the strengths and weaknesses of the involved algorithms.

Parameter setting

Parameter setting or algorithm configuration is the process of optimising the performance of an algorithm by modifying its quantitative parameters. Even though it became a relevant topic in the last decades, its description in many soft computing applications is very rare [259]. This might be related to a lack of well-structured procedures in testing the algorithm performance, as stressed by some authors [118], [260]. The dynamics of the setting procedure allows to distinguish between two strategies: offline parameter tuning (e.g. the setting is a fixed input before the initialization) and online parameter control (e.g. parameters change dynamically along the process) [261]. The procedures discussed hereafter are suitable for offline tuning based on an available training set. Once the parameters are set, the algorithm

can be applied at its best to the problem instance, thus the tuning is an application-dependent task which cannot be avoided to ensure a good DI performance.

Algorithm comparison

The need for test and comparison of different algorithms one against each other on a set of problem instances can be explained in view of the No-Free-Lunch (NFL) theorem, which, roughly speaking, demonstrated that there is no algorithm able to perform better than the others on all the possible problems. On the contrary, if one outperforms on a class of instances, its performance deteriorates on the set of remaining problems. The NFL has been demonstrated both in the field of optimisation and machine learning [262], [263], thus it is relevant for SHM, in general. Since the NFL theorem does not hold for single subsets of problems, it is possible to identify a class of instances in which an algorithm always outperforms the others. Although no univocal acceptance of the NFL theorem exist among scientific community, being based on hypotheses which make it irrelevant in practice according to some authors [264], in some Computer Science fields this theorem led to a proliferation of methods. Nevertheless, innovative contributions hardly arose, being these methods overwhelmed by the chase for novelty and the emphasis on competition rather than comparison. To prevent such a tendency, any study should aim at controlled experimentation instead of competitive testing and should make clear and available in the reports all the aspects for a correct assessment of the process [118]. Following such an approach, in the present study, the novel version of the NSA is the result of an adaptation of existing solutions to the specific characteristics of the SHM field for Civil Engineering structures. Moreover, for each component of the general detection strategy, several existing suitable algorithms are identified and compared on more case studies, to stress the specific advantages and disadvantages. In some situations, it is not possible to find an approach that is always outperforming the others, and this has been pointed out with attention to the reasons why one performs better than the others and the conditions in which this happens. Even when the methodology here developed is compared against other existing computing techniques, the approach is never competitive. Aiming at a proper development of the algorithm, any comparison is made fair by tuning all the alternative methods to find their best instance for each dataset.

3.5.2. Measures of performance

Performance, in the context of Computer Science, can be assessed in terms of quality of the solution, computational effort and robustness [255]. In the present work, the performance is assessed mainly in terms of the quality of the solution, since the main goal is to compare different instances of the same algorithm, thus with very similar robustness and time requirement. Information regarding the

computational burden and the robustness of the methodology is provided anyway. A classification algorithm for damage identification succeeds when it is able to correctly label the system's behaviour. The outputs of the classification (e.g. correct and incorrect assigned labels) in the binary case are four: actual unhealthy labelled as unhealthy (True Positive, TP); actual healthy labelled as healthy (True Negative, TN); actual unhealthy labelled as healthy (False Negative, FN); actual healthy labelled as unhealthy (False Positive, FP). Commonly, they are collected in the so-called confusion matrix $\mathbf{C}$ (Figure 3.8). In terms of error, the false negative class is the most dangerous as it does not warn against a potential fault. On the opposite side, a high number of false positives, even if not dangerous for the structural safety, causes an increase in the cost for the after-warning inspections, reducing drastically the cost-effectiveness of the monitoring.

	ACTUAL UNHEALTHY	ACTUAL HEALTHY
LABELLED UNHEALTHY	TRUE POSITIVE $C_{1,1}$	FALSE POSITIVE $C_{1,2}$
LABELLED HEALTHY	FALSE NEGATIVE $C_{2,1}$	TRUE NEGATIVE $C_{2,2}$

Figure 3.8: Confusion matrix: possible results of the anomaly detection.

In one-class classification, only the false positives can be counted for certain, since no positive samples are available. Therefore, one could be tempted to define a performance metric in terms of FPs and time requirement, expressed in terms of number of generated detectors as:

$$f = a \frac{\text{detectors}}{\text{max.det}} + b \frac{FP}{\text{max.FP}} \quad (\text{Eq. 2.34})$$

where f ranges between 0 and 1 (0 being the best performance), since during the comparison the two parameters are normalised to the highest values found during the experiment (*max.det* and *max.FP*). Moreover, two weighting coefficients are introduced (a and b) to balance the importance of the two terms: for instance, being better to compromise on the memory allocation than on the error reduction, a higher percentage can be assigned to the FPs (75%) and a lower percentage to the number of detectors (25%). The previous metric, timely tested in section 5.2, cannot consider the inverse tendency of FP and FN, thus it might reward a tolerant classifier that, in reality, identifies many positive samples as negative.

In literature, many performance metrics for binary classification have been developed. The most common ones are reported in Table 3.4. For a thorough discussion upon these metrics the interested reader can refer to [265]–[267].

Table 3.4: Typical performance metrics.

Name	Formulation
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN}$
Precision	$\frac{TP}{TP+FP}$
Recall (Sensitivity or True Positive Rate)	$TPR = \frac{TP}{TP+FN}$
False Positive Rate	$FPR = \frac{FP}{TN+FP}$
F-score	$F_{\beta} = \frac{(\beta^2+1)TP}{(\beta^2+1)TP+\beta^2FN+FP}$
Specificity	$\frac{TN}{TN+FP} = 1 - FPR$
Area under the ROC Curve (balanced accuracy)	$ROCAUC = \frac{1}{2}(TPR + (1 - FPR))$
Area under the PR Curve	$PRAUC = precision \cdot TPR + \frac{1}{2} \left(\left(precision + \frac{p}{n+p} \right) (1 - TPR) \right)$
Youden Index	$J = TPR + (1 - FPR) - 1$
Root mean square	$RMSE = \sqrt{\frac{1}{N} \sum (pred(C) - true(C))^2} = \sqrt{\frac{FP+FN}{Tot}}$
Lift	$\frac{\% \text{ positive classified}}{\% \text{ dataset above the threshold}} = \frac{TP/(TP+FN)}{(TP+FP)/(Tot)}$

The two metrics called “Area under the Curve” refer to two different curves commonly used in the field to compare classifiers:

- Reception Operating Characteristic (ROC) curve;
- Precision-Recall (PR) curve.

The former consists in plotting the algorithm performance in a 2-D space whose axes are the False Positive Rate (FPR) and the True Positive Rate (TPR) (Figure 3.9a). Thus, it is possible to visualize the trade-off between true and false alarm rates [268]. ROC demonstrates to be significant for problem instances with unequal class distribution. In the present study, all the algorithm instances are analysed as discrete classifiers, meaning that each of them is a classifier that provides a single point in the ROC space, thus the ROC curve is drawn by connecting the point (0,0) and (1,1) to the algorithm ROC point (FPR, TPR) (Figure 3.9a), and the Area Under the Curve (AUC) is given by the equation in Table 3.4. The closer the value is to 1, the better (1 meaning that 100% of TP are identified with 0% of FP).

The latter graph has been proven to be a good metric for strongly imbalanced samples as well as when the TP identification is much more important and difficult than the TN detection [269]. Like the ROC, it is possible to draw a PR curve for a discrete classifier, namely a classifier which provides a single point in the 2-D space whose axes are the TPR and the precision. In this case, the curve is drawn by connecting

the point $(0, precision)$ and $(1, \frac{P}{N+P})$ to the PR point $(TPR, precision)$ of the algorithm (Figure 3.9b), where P and N indicate the number of nonself and self elements, respectively. Thus, the AUC of the PR is given by the equation in Table 3.4.

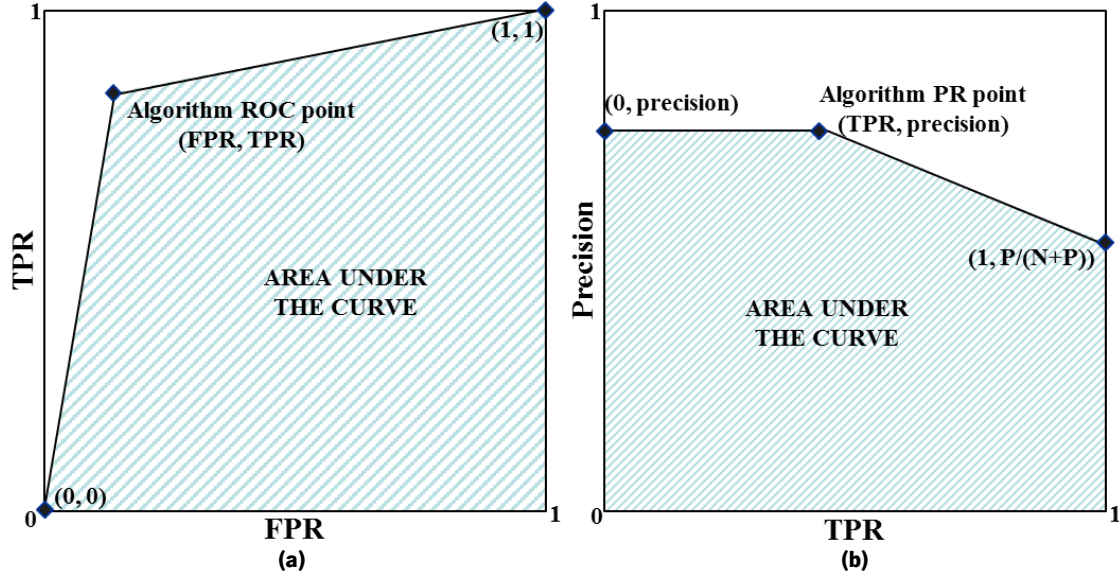


Figure 3.9: Example of performance measures: (a) ROC curve; (b) PR curve.

Many metrics can be derived one from each other, e.g. by linear combination [267] or by combining them in a two-level metric [270]. The F-score, for instance, is the weighted harmonic average of precision and recall, with β expressing the influence of the precision with respect to the recall (common values are $\beta = 2$ or $\beta = 0.5$). The ROC is a linear transformation of the Youden Index. Nonetheless, distinct performance metrics may rank the classifiers in a different order. This is obviously due to the formulation and in particular to their invariance to one or more confusion matrix elements. A metric is invariant to an element of the matrix when a change only in that specific element does not cause any change to the metric, for instance because its formulation is independent of a specific element of the confusion matrix thus, it does not vary when this element varies. The invariance properties of the metrics are studied considering the following 8 possible scenarios of variation in the confusion matrix [265]:

(1) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$	(2) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP & FN \\ FP & TN' \end{bmatrix}$	(3) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP' & FN \\ FP & TN \end{bmatrix}$	(4) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP & FN' \\ FP & TN \end{bmatrix}$
(5) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} TP & FN \\ FP' & TN \end{bmatrix}$	(6) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} k_1 TP & k_1 FN \\ k_1 FP & k_1 TN \end{bmatrix}$	(7) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} k_1 TP & k_2 FN \\ k_1 FP & k_2 TN \end{bmatrix}$	(8) $\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \rightarrow \begin{bmatrix} k_1 TP & k_1 FN \\ k_2 FP & k_2 TN \end{bmatrix}$

(1) exchange of positive and negatives: the metric is invariant to the distribution of classification results, this can be a problem when the data set has different and/or unbalanced class distributions, but it is

preferred when the aim is a measure independent of the performance on a specific class. Accuracy, area under the ROC, RMSE and Youden Index are invariant metrics;

(2) variation of TN: the metric does not assess the ability of the classifier to identify negative samples, it is acceptable when the negative class is multi-modal and defined as “everything not positive”. Non-invariant metrics are preferred when the negative class is well-defined and unimodal. Precision, recall and F-score are invariant metrics;

(3) variation of TP: complementary to the previous scenario. Invariant metrics are reliable for unimodal well-known negative class. Non-invariant measure can be used as stand alone. Specificity and FPR are invariant metrics;

(4) variation of FN: invariant metrics are stable under disagreement between data and negative labels. Non-invariant metrics reward the classifier according to their monotonicity, e.g. if the metric value increases with FN, then classifiers prone to FN errors are favoured. Precision, FPR and specificity are invariant metrics. The other non-invariant metrics decrease as FN increases;

(5) variation of FP: invariant metrics can be reliable against counterexamples, namely outliers in the positive samples which are not explained by the dataset distribution. As in the previous case, non-invariant metrics must be selected according to their monotonicity and the specific problem investigated to favour or disfavour classifiers prone to FP error. As for this case, recall is an invariant metric; the other non-invariant metrics do decrease when FP increases;

(6) uniform increase of dataset: invariant metrics are reliable whenever the change of dataset size is likely to lead to similar proportion of the samples state. The results of a non-invariant metric might depend on the dataset size and it is preferable when no information about how representative of the whole population the samples are. Accuracy, precision, recall, FPR, F-score, specificity, AUC ROC, RMSE, Youden Index and AUC PR are invariant metrics;

(7) changes of positive and negative columns: such change is expected when the additional data is significantly different from the previous dataset, though keeping the same proportion of positive and negative samples. An invariant metric might be unreliable when the performance of the classifier on different data set must be assessed. Precision is an invariant metric;

(8) changes of positive and negative rows: assuming that the proportion of positive and negative samples does not vary, such a change is due to a variation of the algorithm performance within one of the two classes, e.g. if one of the two classes is better represented in the new data. An invariant metric is preferred when no changes in class quality is expected, namely when the positive and negative samples are drawn from the same distribution. Non-invariant metrics are preferred when no information about the quality of

the data samples is available. Recall, FPR, specificity, Youden Index, lift and AUC ROC are invariant metrics.

Aiming at defining the most suitable for the comparison of classifiers in damage detection, a preliminary analysis of the metrics is carried out by considering three artificial cases:

- (1) Same number of positive and negative samples;
- (2) Positive samples > negative samples;
- (3) Positive samples < negative samples.

Table 3.5: Confusion matrices for different classifiers against the same three case studies.

Instance	Case 1				Case 2				Case 3			
	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN
1	100	0	41	59	300	0	41	59	100	0	41	259
2	95	5	39	61	295	5	39	61	95	5	39	261
3	90	10	37	63	290	10	37	63	90	10	37	263
4	85	15	35	65	285	15	35	65	85	15	35	265
5	80	20	33	67	280	20	33	67	80	20	33	267
6	75	25	31	69	275	25	31	69	75	25	31	269
7	70	30	29	71	270	30	29	71	70	30	29	271
8	65	35	27	73	265	35	27	73	65	35	27	273
9	60	40	25	75	260	40	25	75	60	40	25	275
10	55	45	23	77	255	45	23	77	55	45	23	277
11	50	50	21	79	250	50	21	79	50	50	21	279
12	45	55	19	81	245	55	19	81	45	55	19	281
13	40	60	17	83	240	60	17	83	40	60	17	283
14	35	65	15	85	235	65	15	85	35	65	15	285
15	30	70	13	87	230	70	13	87	30	70	13	287
16	25	75	11	89	225	75	11	89	25	75	11	289
17	20	80	9	91	220	80	9	91	20	80	9	291
18	15	85	7	93	215	85	7	93	15	85	7	293
19	10	90	5	95	210	90	5	95	10	90	5	295
20	5	95	3	97	205	95	3	97	5	95	3	297
21	0	100	1	99	200	100	1	99	0	100	1	299

In all three case studies, the number of FP and FN errors does not change. It is worth noting that lift, precision and PR AUC are slightly sensitive to the inverse change in FP and FN. Comparing case 2 and case 1, the larger number of positive samples with the same number of FP leads to an increase of the recall that becomes less sensitive. Consequently, the F-score also reduces in sensitiveness, being an average of recall and precision. Furthermore, it is worth considering that specificity and FPR depend only on TN and FP, thus they cannot appreciate changes in the FN. As well as recall depends only on TP and FN, thus it cannot appreciate changes in the FP. Among the metrics dependent on all the elements of the confusion matrix, ROC AUC and Youden Index are the most sensitive to the changes, while accuracy and RMSE have the same sensitivity due

to their formulation. Case 2 is particularly interesting, since the performances assessed through ROC AUC and accuracy show an opposite trend. This is due to the fact that the accuracy rewards the overall reduction of errors (e.g. if the reduction in FP is higher than the increment in FN or vice versa, the accuracy increases) whereas the ROC AUC first normalises the changes to a class label over the actual elements of that class (e.g. if the reduction in FP is higher than the increment in FN, once the two values are normalised respectively on the total actual positives and the total actual negatives, their relation is inverted, thus the ROC AUC decreases).

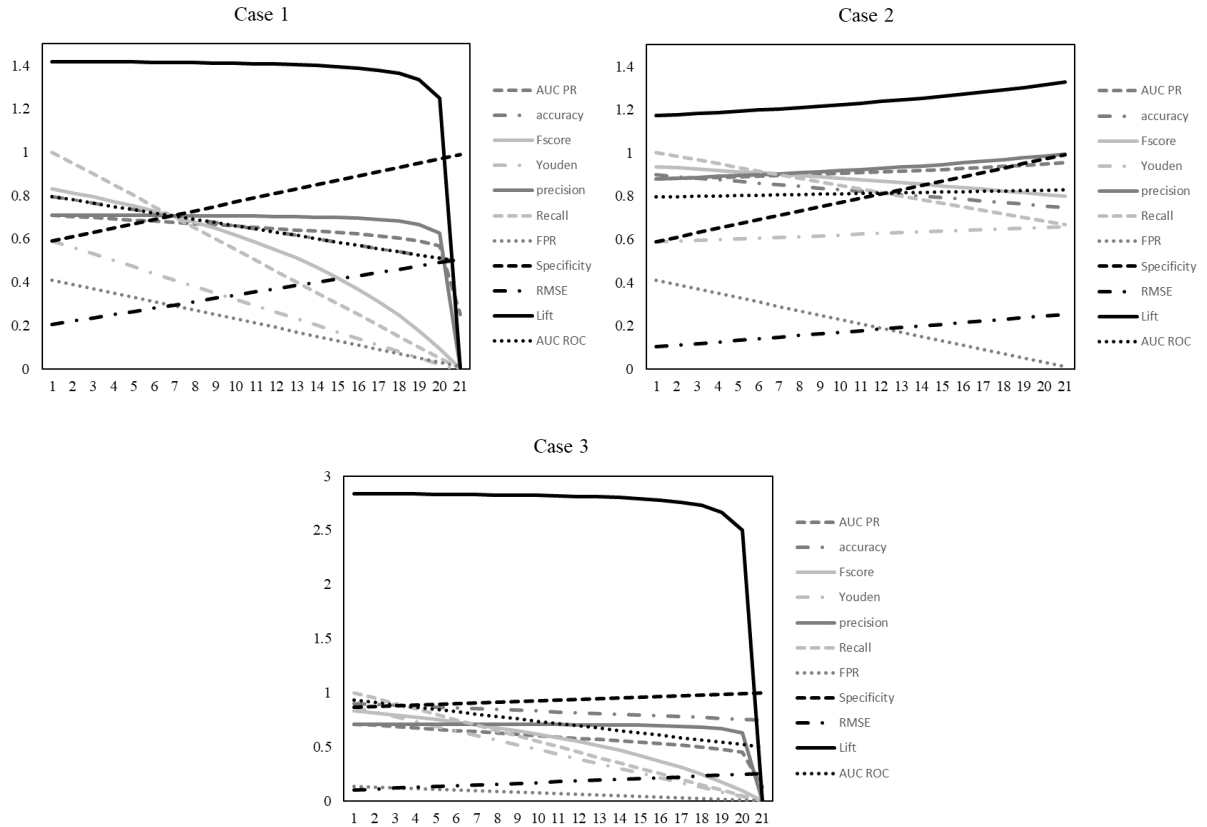


Figure 3.10: Tendency lines of the performance metrics in the analysed case studies.

In conclusion, it is evident that many metrics have been developed with different purposes. A more in-depth analysis of performance metrics is out of the scope of the present work, thus in the applications, the comparison among the classifiers is based only on the ROC AUC since the assessment in terms of normalised number of errors is preferred to the absolute value.

Moreover, it is worth mentioning that the metrics discussed above are used here for a deterministic comparison of the classifier. Usually, the combination of factor levels scoring the highest performance metric is considered the best, thus it is allocated for the monitoring. Under a statistical point of view, such a statement is incorrect or at least unfounded. Indeed, the comparison is carried out over a single dataset which is more or less representative of the actual feature distribution and, if the conclusions are not statistically supported,

one is not allowed to generalise the deterministic conclusions to other populations or to other datasets sampled from the same population. At this stage of the work, it is assumed that being the self sample set a proper approximation of the whole self set, which is consistent for the specific case study, the conclusions are sufficiently reliable to carry out the training and initialise the monitoring. A dynamic formulation of the algorithm enables eventually to update the classifier based on new monitored data, thereby improving the initial set-up. At the present stage of the work, the DI strategy has not been completely cast into a statistical framework, thus, the use of statistical significance and confidence intervals in the design of the methodology for a specific case study is not addressed. However, the stochastic nature of the processes that are investigated in the context of SHM requires at least a reference to statistical analysis and this will constitute a relevant future scope.

3.5.3. Design of the Experiments (DOE)

In experiment design, the common terminology identifies problems as generic classes and problem instances as particular numerical cases. In parameter tuning, algorithm instances consist of different settings of the same algorithm compared against the same problems and problem instances. Finally, in algorithm comparison, different algorithms are compared against the same problems and problem instances. The influence of randomness during the process is assessed repeating, for each design point, more runs, said random trials.

The algorithm proposed in the present work is conceived to avoid any random effect, except for the generation of the datasets when they are produced by polluting real measurements with noise or sampled from a larger population. Nonetheless, even when the selection is completely unbiased, the randomness determines a variability in the results. Due to such variability, even comparing two classifiers whose performance over the whole population is identical, one of the two may outperform the other on a specific randomly drawn dataset. The same can happen considering two algorithms which on the average produce classifiers with the same accuracy, but their performance may be different on a specific randomly drawn training set.

The definition of the algorithm instances for the experiment is based on the design of experiments (DOE), which is a rigorous approach to define the cost-effective sampling (minimum number of runs) to carry out the tests and statistically support the conclusions. In more details, given a process (e.g. the classification) affected by several factors (e.g. algorithm parameters), the DOE supports in the decision of which of them should be changed and how, in order to infer conclusions regarding the effect of the variation. The effect is studied in terms of one or more reliable response variables (e.g. the performance metrics). The factors are analysed as controllable variables which assume in their domains different values called levels [255].

Statistical DOE allows to optimise the ratio between the amount of information that can be obtained and the amount of experimental effort (e.g. time and economical costs). As showed in Table 3.1, the DNSA here developed has 5 quantitative parameters:

- Number of divisions (n) of the side of the unitary space, which is directly related to the detector radius (r_{det});
- Self radius (r_{self});
- Censoring distance (*Cens. dist*);
- Maturity age (t);
- Adaptation rate (η_0).

It is worth noting that the censoring distance, being a linear combination of detector and self radius, is a dependent variable, thus the statistical analysis of its effect on the algorithm performance must take this dependency into account.

Different well-known DOE methodologies are applied to the case studies in the next chapter:

- Full factorial design;
- Central-composite design;
- Response surface method.

The full factorial (FFD) is a common experimental design where all the possible combinations of factor levels are tested. Assuming for instance 2 levels (low or “-” and high or “+”) for 2 factors (x_1 and x_2), the DOE requires $2^2 = 4$ tests of the process, namely (x_1-, x_2-) , (x_1-, x_2+) , (x_1+, x_2-) and (x_1+, x_2+) . In the present work, three-level full factorial design is followed, meaning that three levels are predefined per each factor (e.g. low, medium and high level or “-1”, “0” and “1”), considering 2 or 3 factors. This means that 3^2 or 3^3 experiments are carried out. The three-level design improves the simple two-level by allowing to investigate nonlinearity and curvature in the response function, namely the quadratic effect due to each individual factor.

The Box-Wilson central composite design (CCD) is composed of an embedded factorial or fractional factorial design with centre points and a set of so-called “star points” to analyse the curvature. In a scale where the distance between the centre of the design space and the factors of the embedded factorial design is equal to 1, the star points are located to a distance from the centre of the space equal to $\alpha = [n^\circ \text{ factorial runs}]^{1/4} > 1$. Other values of α can be set, but this one allows to guarantee the rotatability. The variance of the predicted value for a generic point in the design space, using a rotatable design, is a function of the distance from the point to the centre of the design space and it is independent of the direction from the centre to such point. A non-rotatable design induces a bias in the investigation

of the optimum response, since the direction from the centre to the points affects the variability of the prediction. FFD is not rotatable, thus the information function of the model, namely $\frac{1}{V(\hat{y})}$ where $V(\hat{y})$ is the variance of the predicted value, is not a simple convex function with a peak in the centre of the design space and a value that depends only on the distance from the centre itself, but depends also on the distribution of the points with respect to the centre. Comparing CCD and FFD, two configurations are possible:

- Circumscribed, when the embedded factorial design of CCD is equal to the FFD and the star points are new low and high extremes;
- Inscribed, when the star points of CCD are as far as the low and high levels of the FFD from the centre of the design space, thus the embedded factorial design of CCD is scaled down with respect to the FFD.

In the present study, the CCD is defined for two factors and eventually repeated for the three levels of a third factor. Considering an embedded two-level factorial design, the star points are placed at $\alpha = [2^2]^{1/4} = 1.414$. If an inscribed CCD is performed (Figure 3.11), therefore, the star points are located at the maximum and the minimum levels of the full factorial design. This is essentially done to compare the two approaches in terms of achieved response surfaces, without extending the limit of the levels to avoid too large or too small radii.

Considering an FFD with 3 levels for 3 factors, this means that 27 tests are performed. For the same case study, the CCD requires 5 levels – the centre point, two factorial points and two star points per each factor, but still 27 combinations (considering 9 combinations per each one of the three levels of the third factor). The full factorial design, in fact, can be interpreted as a central composite design where the star points are at the centre of the faces of the factorial space, but no rotatability exists in this case. Due to the deterministic generation of detectors based on an integer number of side divisions, the levels are not exactly scaled as in the theoretical design (-1.414, -1, 0, 1, 1.414), but as close as possible to such values.

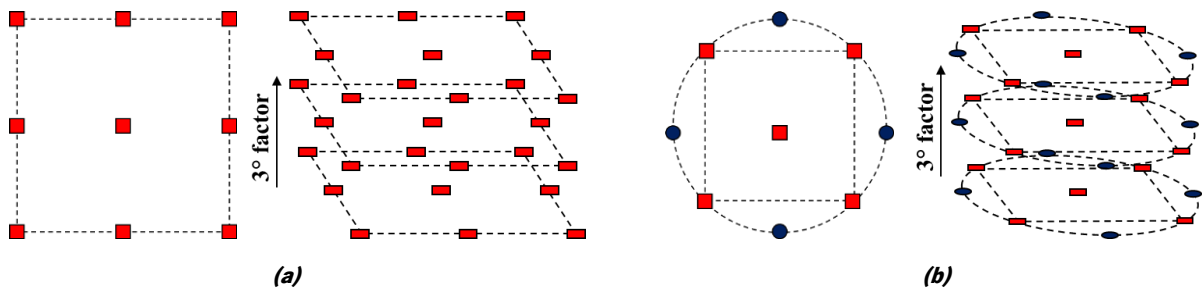


Figure 3.11: Comparison between full factorial design (a) and central composite design (b). The red squares are the levels of the factorial design and the blue circles are the star points. Three and two factor designs are shown.

When the response function is investigated simply based on the FFD or the CCD, the optimal combination of factors levels is selected only within the samples, as the one with highest overall performance metric. Finally, the response surface method refers to the definition of a polynomial approximation of the response function by curve fitting and regression. The response surface is a function of the factors and their interactions, also considering, if needed, higher order terms (e.g. quadratic or cubic).

In the present study, the response surface is estimated as function of two factors x_1 and x_2 , according to two possible models:

$$\begin{aligned}\hat{y} &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 \\ \hat{y} &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \beta_6 x_1^2 x_2 + \beta_7 x_1 x_2^2 \\ &\quad + \beta_8 x_1^3 + \beta_9 x_2^3\end{aligned}\quad (\text{Eq. 3.35})$$

where the former is a quadratic model and the latter a cubic model. In both the cases, the complete model can be reduced to remove the terms which are not significant. The definition of the response surface follows the two aforementioned DOEs. In fact, the function is fitted by regression analysis based on the treatment samples. Such samples can be defined according to the FFD or the CCD. Then, the response surface is analysed as an approximation of the response function to find the optimal combination of factors levels not only among the samples. Due to the deterministic generation of the detectors based on an integer number of side divisions, the levels are not exactly scaled as in the theoretical design $(-1.414, -1, 0, 1, 1.414)$, thus the variables are not coded to the space centred in 0 before the regression analysis and the original value is kept.

Each problem instance used to test a classification algorithm is a population set D from which a subset of known pairs $\langle \mathbf{x}, f^{-1}(\mathbf{x}) \rangle$ are drawn randomly and subdivided into the mutually exclusive pilot study set P , training set R , validation set V , and testing set T (Figure 3.12). The exploratory or pilot study is a preliminary sensitivity analysis on only a few representative samples which helps set the factors levels, fix the parameters with minimal effect and connect the factors whose effect is strongly correlated. Pilot studies also provide information on outcomes distribution, better performance measures and number of runs for any instance based on variability due to randomness [256]. The learning algorithm, then, uses the training data to generate the classifier g . After the development of several classifiers with different parameter values, the validating set V is used to estimate the performance and find the best setting. Finally, the testing set T is used to assess the performance on new data, eventually comparing the performance of different learning algorithms [1], [271]. For one-class classification algorithm the training set is composed of normal samples only. However, as already mentioned, it is important to include also abnormal samples in the validating set to penalise tolerant classifiers. Any algorithm setting must be done

before seeing the testing data. Statistical testing for algorithm comparison is, thus, performed by defining three datasets, namely a training, a validating and a testing set, and comparing different algorithms only on the last, once optimised on the former two [258].

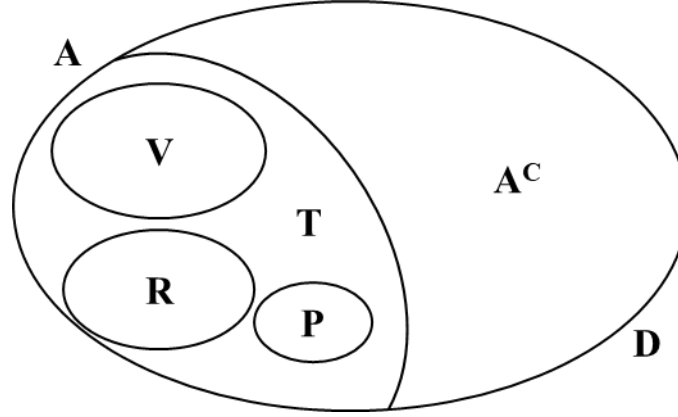


Figure 3.12: Problem set for pilot study (P), training (R), validation (V) and testing (T) as subset of analyses samples (A) which is also a subset of the entire problem set (D). T is the complementary of the union of V , R and P sets.

For parameter setting, different plots of the performance metric(s) can be exploited to show the influence of the different factor levels. Four main graphs can be used:

- Estimated Marginal Means (EMM) plot: mean of the performance metric values averaged across all the levels of the factors – it allows a first definition of the factor effect and to define the best level of the single factor, independently of the others. For instance, Figure 3.13a reports the EMM plot for three factors at three levels, each point is the mean of the tests carried out where the factor was at that level. When the metric increases with the performance quality, the factor level with higher mean value is the best for the single factor. Moreover, the difference between the mean values provides information regarding the variability of the performance depending on the specific factor, namely, which factor affects the most the algorithm performance. In the instance of Figure 3.13a, the performance is independent of factor 3, whereas the best performance is provided by the intermediate value for factor 1 and 2. Moreover, the performance depends non-linearly on these two factors as it increases when the two factors grow from the low to the intermediate level but then drops as they grow to the high level.
- Estimated Standard Deviations (ESD) plot: mean of the standard deviation of the performance metric values for different repetitions of the test, averaged across all levels of the factors, as for the EMM – it allows to estimate the effect of randomness on the factors. A lower variability means more robustness. For instance, Figure 3.13b reports the ESD plot for the same three factors at three levels over 6 repetitions.

- Difference between Marginal Means (DEMM) plot: difference of the marginal mean values averaged at different levels per each factor – it allows to define the significance of a certain variable, namely if it affects significantly the performance metric. For instance, Figure 3.13c reports the differences of the previous case. The labels of the x axis refer to the levels compared (e.g. “-1 vs 1” refers to the difference between the mean at high and low level). It appears clearly that, for factor 1 and 2, the difference in the performance changing from intermediate to high level is the largest.
- Interaction plots (IPs): marginal means for different values representing the interaction between two factors – it allows to extend the information of the EMM plot by considering the interaction of the different levels of two factors per time. For instance, Figure 3.14 shows the mean per each level of a three-level, three-factor full factorial design. In this graph, overlapping or parallel lines show a low interaction, whereas intersections show an interaction of the factors. Moreover, it allows to define the best combination of levels two-by-two. Whenever there is interaction the definition of the best factor combination must be carefully analysed since the performance trend due to the variation of a factor level depends on the level of the interacting factor. For instance, in Figure 3.14a, the performance metric increases with the factor 1 when the factor 2 takes its low level, but decreases with increasing values of factor 1 when the factor 2 takes the high level.

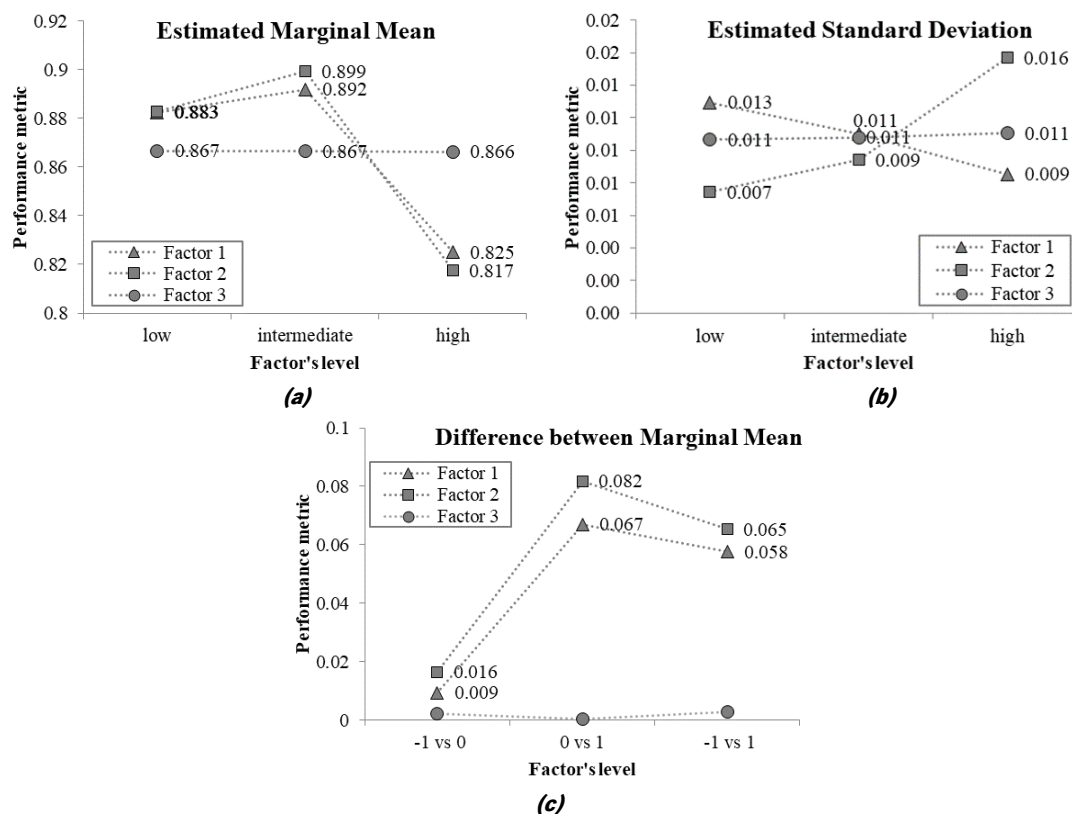


Figure 3.13: (a) Estimated marginal means, (b) differences between the means and (c) standard deviations for the three independent variables.

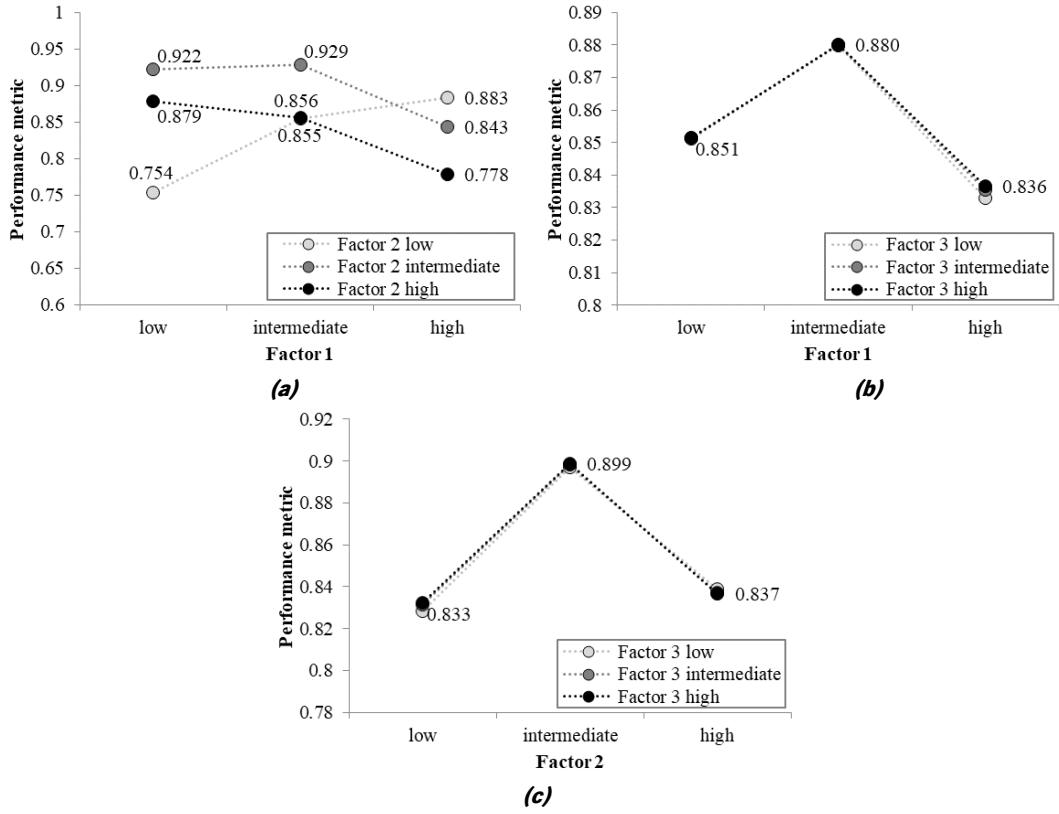


Figure 3.14: Interaction plots combining three factors: (a) factor 1 and 2; (b) factor 1 and 3; (c) factor 2 and 3.

3.5.4. Selection of case studies

In algorithm design and testing, the selection of the problem instances aims at exploring the problem domain to assess the sensitivity of the algorithm parameters and to stress strengths and weaknesses that require improvement. Different sources of problem instances exist:

- Real world datasets;
- Random variants of real datasets;
- Artificial datasets;
- Published and online libraries;
- Random generation.

Algorithms tested against real data sets do not suffer problems of scalability, being already assessed at the real-world scale, but available real data sets are only a few and representative of actual or past time instances, so maybe outdated. Furthermore, real-world problems are expensive to collect and sometimes are not commonly shared among researchers. Generalization of the inference done on the problem is questionable since the population of problems from which they are drawn is not usually known [272]. Random changes to real data preserving only the original structure allow to slightly increase the number

of problems and their size. Artificial datasets differ from the previous since they are generated by the analysis of a purely numerical model not always representative of a real case study. Published libraries are composed of real or artificial data sets commonly used as benchmark instances. It is common to use them to compare or develop algorithms, even though sometimes benchmarks have only the advantage of the knowledge of the optimal solution. They are often not sufficiently representative of real-world instances, being rather simple and small. Consequently, almost every algorithm shows excellent performances despite its complexity, nature and quality, but this cannot support any generalization to the vast realm of real-world applications. It is also worth noting that benchmark problems introduce a bias into the field of research since, whenever a new algorithm outperforms a previous one on such problems, it is considered worth publishing, generating a series of methodologies extremely efficient on a very limited problem class, whereas the development of different methods and the understanding of their real behaviour on different classes of problems are neglected [118]. In fact, any inference made on the basis of the tests applies only to the data domain from which the dataset is sampled or to the problem population in case multiple domains are considered [258]. The generalisation of the inference depends on how representative the analysed domains are of other classification problems; a repetitive use of the same benchmarks ultimately leads to algorithms that are tailored to specific datasets. Completely random-generated instances could overcome the drawbacks of the previous sources, providing an extreme variability in the problem population and avoiding limitation in their number and size [256]. Although this type of problems may not resemble real problems, they can stimulate the algorithms by stressing on their or their components weak points [118]. In conclusion, a proper selection of the problem instances should include numerical and real-world problems together. This is the research strategy that has been followed in the present thesis. Among the datasets analysed during the algorithm development, four main case studies are presented and discussed in depth in the next two chapters:

- Concrete frame building – numerical model, artificial dataset non-calibrated to a real case study;
- I40 bridge over Rio Grande – numerical model, artificial dataset calibrated to the real case study;
- Segmental masonry arch – laboratory test, real dataset in controlled conditions;
- Church of Monastery of Jerónimos – field data acquisition, real dataset from a large complex structure.

The numerical datasets are analysed in chapter 4 whereas the experimental datasets in chapter 5. This organisation divides the case studies according to the nature of the datasets, following the increasing complexity of the problem. However, they have not been analysed according to the aforementioned sequence as it may result evident from the discussion in the related sections. In more detail, the algorithm

was first developed and its performance in the classification was verified by employing the data from the Monastery of Jerónimos (Figure 3.15). The available dataset was composed of many samples, but only belonging to a condition considered healthy. Thus, the methodology to simulate artificial nonself data, by generating outliers, and the methodology to approximate the self space have been developed and tested with this case study. The DI strategy showed promising results, nonetheless the lack of data belonging to a real damaged state did not allowed a proper validation. This was achieved by analysing two artificial case studies, where a full control over all the testing conditions was possible. First, the concrete frame building was used for a complete validation of the algorithm features by using the natural frequencies as damage-sensitive parameters; then, the I40 bridge was employed to extend the algorithm validation when resorting to signature analysis. I40 bridge is a case study closer to reality due to the initial calibration of the model through the field data collected while progressively damaging the structure until collapse. The type of damage analysed in this case is a typical expected scenario for this kind of bridges; moreover, the data used for the damage detection are obtained by simulating a structural monitoring through accelerometers. Lastly, a final test was necessary to assess the algorithm effectiveness against field testing data. Therefore, the mock-up of a segmental masonry arch was exploited to obtain data from a real structure subjected to realistic damage scenarios.

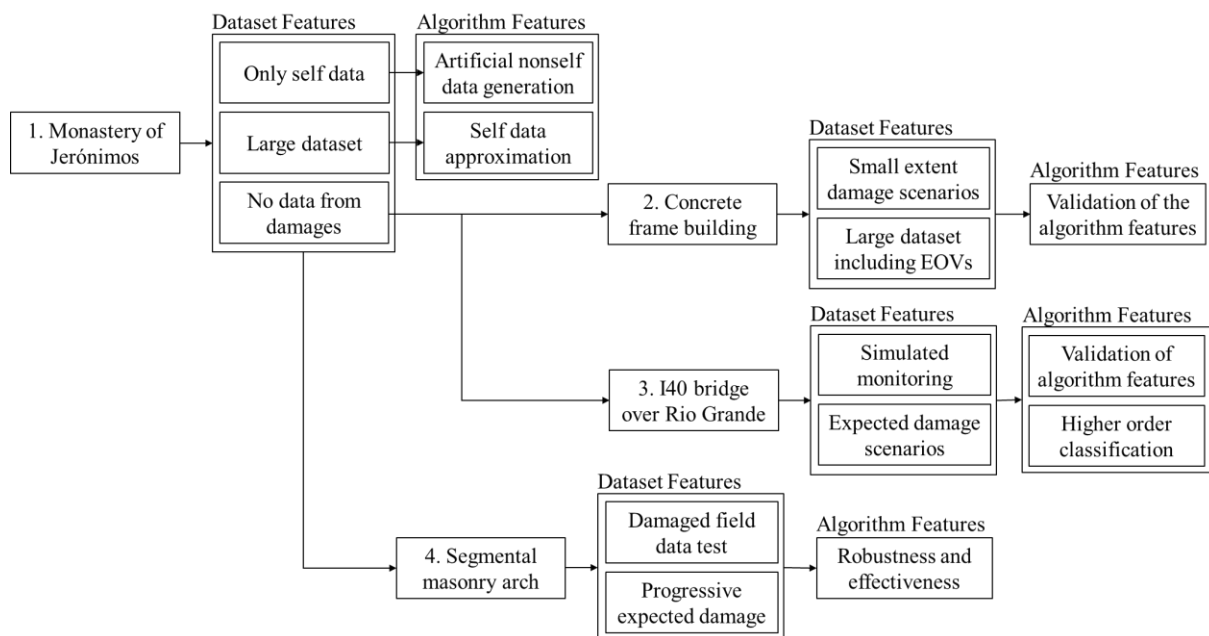


Figure 3.15: Workflow of algorithm validation and testing.

3.6. Conclusion

The first part of this chapter described the damage detection strategy that constitutes the main core of the present thesis, and all its main features are analysed. Three main aspects were identified and discussed: (1) the definition of a specific algorithm instance based on a newly developed version of the NSA with deterministic generation of detectors as well as on a simple but effective strategy to reduce the false alarm rate by using two counters of the outputs; (2) the definition of the problem instance by choosing the features to be used for the classification, by selecting a proper number of self samples for the training and by generating nonself samples for validation in case they are not available; (3) the extension of the methodology to higher identification levels by means of two possible supervised learning strategies. All the components of the global damage detection methodology were described in detail.

The second part of the chapter was dedicated to the algorithm testing, which is an aspect of utmost importance for a correct inference about the methodology's performance. Numerical experiments were dissected in their main components, namely the goals, the performance metrics, the statistical design adopted and the selection of the case studies. This part outlined the fundamentals necessary to the comprehension and discussion of the main results in the next two chapters.

CHAPTER 4

Application and Validation through Different Case Studies

Abstract

This chapter presents the application of the developed methodology to two numerical case studies: a three-story concrete frame building and a three-span steel and concrete bridge. Artificial case studies allow a full control over all the parameters that might affect the structural response and the performance of the damage detection. In more detail, the concrete frame building, being an ad-hoc developed instance, is used for a complete validation of the algorithm features. The I-40 bridge model, being calibrated to a real-world well-known benchmark case, is an interesting testbed for the detection strategy which combines the high control, typical of numerical simulations, with the representativeness of real-world instances. The actual bridge was intentionally damaged before its demolition and the scenarios are reproduced in the numerical model. They are typical damages expected for this kind of bridges; moreover, the data used for the damage detection are obtained by simulating a structural monitoring through accelerometers, allowing to extend the algorithm validation when resorting to signature analysis.

4.1. Introduction

The development of the damage detection strategy based on the DGNSA, as described in chapter 3, requires a long stage of pilot testing and validation, in order to analyse the modified version of the classification algorithm and all the other features of the methodology. Numerical case studies, namely artificial datasets produced by using Finite Element (FE) models, are extremely beneficial in this stage. Indeed, they provide a complete control on all the endogenous and exogenous factors of the structural system and allows to simulate a large number of damage scenarios that in real-world applications would be too expensive and unpractical to achieve. On the other hand, such case studies often lack the sources of uncertainty and the complexity of real-world case studies that must be considered in a following stage of the algorithm testing. Chapter 5 is entirely devoted to such real-world applications, whereas the present chapter collects the main results achieved during the validation stage. Two main case studies are discussed hereafter. The first is a hypothetical concrete frame building. In this case, a low complexity, both of the structure and of the damage modelling strategy, is valuable as it allows to produce an extremely large - yet easy to generate and process - database, including different damage scenarios and sources of variability, as the temperature gradient and the imposed load fluctuation. Such a database is used to analyse the main features of the damage detection methodology. Most of such features are further tested considering other case studies that are more similar to real-world applications. Therefore, only the most relevant results of this stage are presented. The second case study, instead, is not fictitious, as it consists of a FE model, calibrated using the experimental data from the I-40 bridge over Rio Grande in New Mexico. This is a well-known benchmark in the SHM field: a bridge tested with controlled induced damage before its demolition. Finally, it is worth mentioning that the lack of complexity of the artificial cases enables to extend the methodology to higher order classification, namely to identify the location and the extent of the damage, out of a list of previously known scenarios, according to a supervised learning approach. Two ways of achieving this are explored, one based on multiple classifiers in parallel, tested on the case of the concrete frame building, and the other one based on the use of signatures as damage sensitive features, tested on the case of the I-40 bridge.

4.2. Concrete frame building

4.2.1. Description of the case study

A large artificial dataset is generated simulating a concrete building. The building is a three-story structure with rectangular $12 \times 12 \text{ m}^2$ plan and twelve pillars per floor (Figure 4.1). The QoI, namely the artificially

generated natural frequencies, are extracted from the model by solving the eigenvalue analysis. All the properties of the building are considered deterministic excluding the Young's modulus of the pillars and the mass of the slabs.

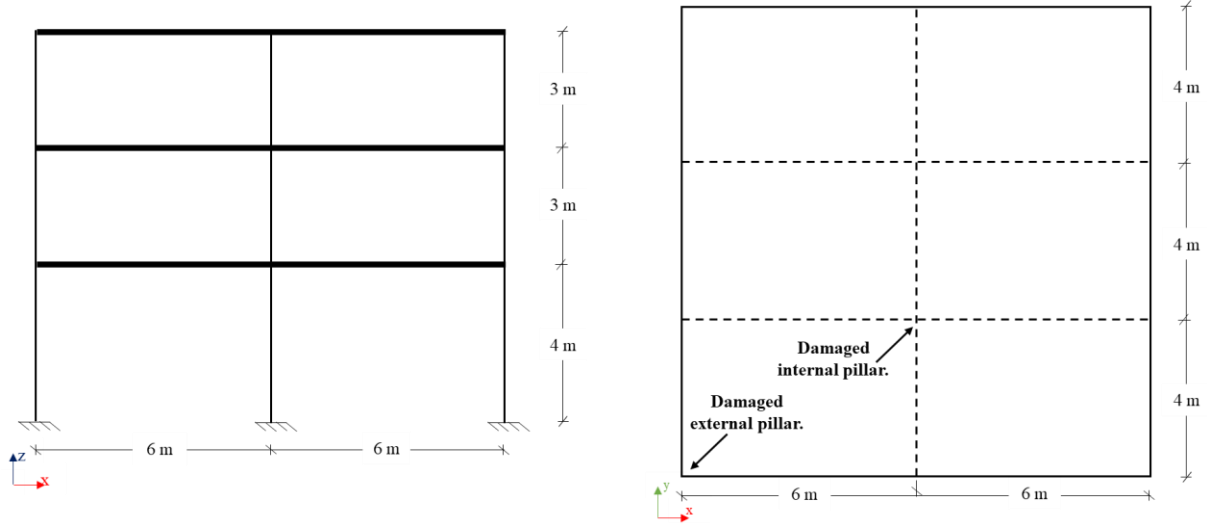


Figure 4.1: Three-story concrete building: on the left, schematic vertical section; on the right, horizontal plan view.

The mass is a stochastic variable, defined as the sum of the deterministic self-weight of the structural and non-structural components, plus a live load formulated according to the JCSS probabilistic model code [273]. Namely, assuming that the building is occupied by offices and considering only the long-term imposed load component, the live load follows a Gumbel distribution with mean $\mu_{5y} = 0.5 \text{ kN/m}^2$ and

standard deviation $\sigma = \sqrt{\sigma_V^2 + \sigma_U^2 \frac{A_0}{A}} k = 0.436 \text{ kN/m}^2$, where the standard deviations of the distributions U and V are $\sigma_V = 0.3 \text{ kN/m}^2$ and $\sigma_U = 0.6 \text{ kN/m}^2$, respectively; the reference area $A_0 = 20 \text{ m}^2$, the influence area $A = 144 \text{ m}^2$ and the shape of influence line factor $k = 2$. Since this formulation has five years of renewal time, the mean of the imposed load for fifty years of design life is calculated as: $\mu_{50y} = \mu_{5y} + 0.78\sigma \ln(10) = 1.28 \text{ kN/m}^2$. The uncertain behaviour of the mass is modelled to consider the possible fluctuation in the permanent load due to the live load in different repetitions of a measure along the time. The self-weight is kept constant, assuming that the building does not change along the monitoring time, and is equal to a mass of 0.9 ton/m^2 for the first floor, 0.75 ton/m^2 for the second and 0.6 ton/m^2 for the last floor.

The Young's modulus is function of the external temperature, according to the following formulation proposed in the Model Code [274]:

$$E = E_{20}(1.06 - 0.003 T) [\text{kN/m}^2] \quad (\text{Eq. 4.36})$$

where $E_{20} = 30000000 \text{ kN/m}^2$ is the Young's modulus at 20°C .

This numerical case study allows to carry out all the preliminary tests on the algorithm. In more detail, the aim is to compare different features, different algorithm parameter values, different designs of the experiments and different performance metrics. The sensitivity of the numerical strategy to such factors is analysed considering changing operational and environmental conditions, as in a real-world application. The limits of the algorithm are explored, considering small damage extents.

The dataset is composed of samples belonging to the undamaged state and 6 damage scenarios, considering, per each scenario, 1005 random samples (viz. 5 samples per each value of the temperature in the range $10\text{-}30^\circ\text{C}$, with a step of 0.1°C). The scenarios analysed are reported in Table 4.1.

Table 4.1: Analysed damage scenarios, with indication of damage extent and location and of the number of samples for the training and the monitoring phases.

Scenario	Pillar	Floor	Extent	Training samples	Monitoring samples
D0	–	–	–	402	603
D1	Internal	1	50%	201	804
D2	Internal	2	50%	201	804
D3	Internal	3	50%	201	804
D4	External	1	50%	201	804
D5	External	2	50%	201	804
D6	External	3	50%	201	804

4.2.2. Testing campaign

The finite element model of the concrete frame building is used to analyse all the possible factors taken into account in the present study. The analysis is carried out considering the following aspects, divided according to the tasks order for the application of the DI strategy based on DGNSA, as presented in chapter 3:

- 1.1) Data scaling: three algorithms (Min-Max Normalisation, Soft Max Scaling and Z-Score);
- 1.2) Censoring strategy: single approach (deterministic generation);
- 2.1) Monitored features: several combinations of natural frequencies, difference between frequencies and temperature values;
- 2.2) Self approximation: several approaches (full self samples set, deterministic approximated self space, affinity method approximation, Gaussian method approximation and B-NN-ASR approximation);
- 2.3) Nonself samples for training: two approaches (a set of real nonself samples and simulated nonself samples);

- 3.1) Algorithm parameters: calibration of detector radius, self radius, overlapping distance for training, adaptation rate and maturity age;
- 3.2) Design of the experiments: three different approaches (full-factorial design, central composite design and response surface).

To avoid repetitions in the present work, only the most important results are described and discussed hereafter, in particular as concerns some preliminary conclusions that are assumed as valid in the next case studies. Other aspects, as the comparison of different designs of experiments, or the simulation of self or nonself samples are analysed in more details in the next applications.

Therefore, out of all the performed analyses, only the combinations reported in Table 4.2 are further described. First, 15 analyses are performed using all the self and nonself samples for the training. The influence of the data scaling algorithm and the damage sensitivity of the features are compared. The comparisons among different instances are summarised in Table 4.3.

Table 4.2: List of analyses carried out.

Code	Scaling	QoI	Self Approx.	Nonself Data	DoE	Det.	Rep.
T1.1	MMN	Distance(3-2,3-1)	None	Damaged	FF	–	4
T1.2	MMN	1 st freq.-2 nd freq.	None	Damaged	FF	–	4
T1.3	MMN	1 st freq.-3 rd freq.	None	Damaged	FF	–	4
T1.4	MMN	2 nd freq.-3 rd freq.	None	Damaged	FF	–	4
T1.5	MMN	Temp-1 st freq.	None	Damaged	FF	–	1
T1.6	MMN	Temp-2 nd freq.	None	Damaged	FF	–	1
T1.7	MMN	Temp-3 rd freq.	None	Damaged	FF	–	1
T2.1	SMS	Dist. (3-2, 3-1)	None	Damaged	FF	–	1
T2.2	SMS	1 st freq.-2 nd freq.	None	Damaged	FF	–	1
T2.3	SMS	1 st freq.-3 rd freq.	None	Damaged	FF	–	1
T2.4	SMS	2 nd freq.-3 rd freq.	None	Damaged	FF	–	1
T3.1	ZS	Dist. (3-2, 3-1)	None	Damaged	FF	–	1
T3.2	ZS	1 st freq.-2 nd freq.	None	Damaged	FF	–	1
T3.3	ZS	1 st freq.-3 rd freq.	None	Damaged	FF	–	1
T3.4	ZS	2 nd freq.-3 rd freq.	None	Damaged	FF	–	1

Table 4.3: List of the factor comparisons.

Code	Goal
T1.1-T1.2-T1.3-T1.4-T1.5-T1.6-T1.7; T2.1-T2.2-T2.3-T2.4; T3.1-T3.2-T3.3-T3.4	Compare damage sensitivity of the features.
T1.1-T2.1-T3.1; T1.2-T2.2-T3.2; T1.3-T2.3-T3.3; T1.4-T2.4-T3.4	Compare the influence of the data coding algorithms.
T1.2-T1.3-T1.4; T2.2-T2.3-T2.4; T3.2-T3.3-T3.4	Compare the influence of the monitored features.

A full factorial design of the experiments is carried out. Thus, per each case, 27 sets of detectors are generated and matched with the self and nonself samples. Each algorithm instance differs for the algorithm parameters. Considering a three levels factorial design, the factors analysed are the detector radius, the self radius and the overlapping distance according to the combination reported in Table 4.4.

Table 4.4: Combination of level for the full factorial design.

Comb.	r_{det}		r_{self}	Censoring distance	
	N° divisions	Value		Formula	Value
1	70	0.0101	0.01	r_{det}	0.010102
2	70	0.0101	0.01	$0.5(r_{det} + r_{self})$	0.010051
3	70	0.0101	0.01	$(r_{det} + r_{self})$	0.020102
4	70	0.0101	0.035	r_{det}	0.010102
5	70	0.0101	0.035	$0.5(r_{det} + r_{self})$	0.022551
6	70	0.0101	0.035	$(r_{det} + r_{self})$	0.045102
7	70	0.0101	0.07	r_{det}	0.010102
8	70	0.0101	0.07	$0.5(r_{det} + r_{self})$	0.040051
9	70	0.0101	0.07	$(r_{det} + r_{self})$	0.080102
10	20	0.0354	0.01	r_{det}	0.035355
11	20	0.0354	0.01	$0.5(r_{det} + r_{self})$	0.022678
12	20	0.0354	0.01	$(r_{det} + r_{self})$	0.045355
13	20	0.0354	0.035	r_{det}	0.035355
14	20	0.0354	0.035	$0.5(r_{det} + r_{self})$	0.035178
15	20	0.0354	0.035	$(r_{det} + r_{self})$	0.070355
16	20	0.0354	0.07	r_{det}	0.035355
17	20	0.0354	0.07	$0.5(r_{det} + r_{self})$	0.052678
18	20	0.0354	0.07	$(r_{det} + r_{self})$	0.105355
19	10	0.0707	0.01	r_{det}	0.070711
20	10	0.0707	0.01	$0.5(r_{det} + r_{self})$	0.040355
21	10	0.0707	0.01	$(r_{det} + r_{self})$	0.080711
22	10	0.0707	0.035	r_{det}	0.070711
23	10	0.0707	0.035	$0.5(r_{det} + r_{self})$	0.052855
24	10	0.0707	0.035	$(r_{det} + r_{self})$	0.105711
25	10	0.0707	0.07	r_{det}	0.070711
26	10	0.0707	0.07	$0.5(r_{det} + r_{self})$	0.070355
27	10	0.0707	0.07	$(r_{det} + r_{self})$	0.140711

4.2.3. Results

A preliminary study concerns the influence of maturity age t and adaptation rate η_0 on the algorithm performance. In this regard, analyses T1.1 to T1.7 are repeated, changing the value of $t \in [15; 30; 45]$ and $\eta_0 \in [0.0025; 0.005; 0.0075]$. The average and the standard deviation of the ROC AUC over the repetition is reported in Table 4.5.

Table 4.5: Statistics of the ROC AUC with varying maturity age and adaptation rate.

	T1.1		T1.2		T1.3		T1.4		T1.5		T1.6		T1.7	
	mean	st dev	mean	st dev	mean	st dev	mean	st dev	mean	st dev	mean	st dev	mean	st dev
	0.788	0.020	0.605	0.042	0.818	0.011	0.881	0.018	0.755	0.001	0.752	0.003	0.749	0.001
	0.788	0.020	0.605	0.041	0.817	0.011	0.880	0.018	0.747	0.002	0.746	0.003	0.747	0.001
	0.748	0.005	0.506	0.002	0.690	0.008	0.691	0.010	0.586	0.001	0.584	0.002	0.574	0.001
	0.788	0.020	0.605	0.042	0.818	0.011	0.881	0.018	0.755	0.001	0.752	0.003	0.749	0.001
	0.731	0.008	0.505	0.002	0.659	0.006	0.658	0.008	0.569	0.001	0.566	0.001	0.559	0.001
	0.585	0.004	0.504	0.002	0.503	0.000	0.503	0.001	0.509	0.000	0.509	0.000	0.508	0.000
	0.788	0.020	0.605	0.042	0.818	0.011	0.881	0.018	0.755	0.001	0.752	0.003	0.749	0.001
	0.605	0.000	0.504	0.002	0.503	0.000	0.503	0.001	0.515	0.001	0.515	0.001	0.512	0.001
	0.515	0.000	0.504	0.002	0.502	0.000	0.502	0.001	0.502	0.000	0.502	0.000	0.502	0.000
	0.764	0.008	0.541	0.024	0.732	0.011	0.798	0.014	0.550	0.002	0.547	0.001	0.538	0.001
	0.474	0.032	0.536	0.007	0.500	0.011	0.506	0.005	0.537	0.001	0.532	0.002	0.522	0.001
	0.683	0.007	0.503	0.002	0.615	0.015	0.625	0.009	0.529	0.001	0.527	0.001	0.516	0.001
	0.764	0.008	0.541	0.024	0.732	0.011	0.798	0.014	0.550	0.002	0.547	0.001	0.538	0.001
	0.765	0.008	0.569	0.033	0.732	0.010	0.801	0.014	0.549	0.002	0.545	0.001	0.537	0.002
	0.572	0.003	0.503	0.002	0.502	0.000	0.502	0.000	0.504	0.000	0.505	0.000	0.505	0.000
	0.764	0.008	0.541	0.024	0.732	0.011	0.798	0.014	0.550	0.002	0.547	0.001	0.538	0.001
	0.638	0.006	0.503	0.002	0.541	0.008	0.543	0.006	0.517	0.000	0.514	0.000	0.514	0.001
	0.525	0.003	0.503	0.002	0.502	0.000	0.502	0.000	0.501	0.000	0.501	0.000	0.501	0.000
	0.675	0.009	0.537	0.025	0.656	0.007	0.686	0.024	0.521	0.001	0.520	0.001	0.511	0.001
	0.437	0.010	0.531	0.005	0.507	0.003	0.555	0.001	0.528	0.002	0.524	0.001	0.510	0.002
	0.630	0.012	0.510	0.012	0.581	0.006	0.579	0.008	0.516	0.001	0.516	0.001	0.509	0.000
	0.675	0.009	0.537	0.025	0.656	0.007	0.686	0.024	0.521	0.001	0.520	0.001	0.511	0.001
	0.451	0.003	0.544	0.009	0.503	0.004	0.557	0.003	0.526	0.003	0.522	0.003	0.511	0.003
	0.553	0.011	0.510	0.012	0.502	0.001	0.502	0.001	0.505	0.000	0.504	0.000	0.504	0.000
	0.675	0.009	0.537	0.025	0.656	0.007	0.686	0.024	0.521	0.001	0.520	0.001	0.511	0.001
	0.675	0.009	0.532	0.016	0.649	0.019	0.687	0.025	0.520	0.001	0.519	0.001	0.509	0.001
	0.516	0.010	0.510	0.012	0.501	0.001	0.501	0.001	0.502	0.001	0.502	0.001	0.502	0.001
Max	0.788	0.032	0.605	0.042	0.818	0.019	0.881	0.025	0.755	0.003	0.752	0.003	0.749	0.003
Min.	0.437	0.000	0.503	0.002	0.500	0.000	0.501	0.000	0.501	0.000	0.501	0.000	0.501	0.000
Avg.	0.651	0.010	0.534	0.016	0.627	0.007	0.655	0.010	0.561	0.001	0.559	0.001	0.553	0.001

Table 4.6 reports the mean and standard deviation for different r_{det} , r_{self} and censoring distance, being constant the other parameters, in a single repetition of each instance. Comparing this with the results summarized in Table 4.5, it is possible to see that the standard deviation induced by maturity age and adaptation rate on the average is 0.001 in most of the instances (maximum 0.016 in instance T1.2) whereas, for the other parameters, it is at least seven times larger for all the instances, excluding T1.2 where it is almost coincident. In this instance, however, the ROC AUC is generally lower than in the other cases and its variability is lower.

Table 4.6: Examples of the variability induced by the three main parameters on the ROC AUC.

	r_{det}		r_{self}		Cens. Dist.			r_{det}		r_{self}		Cens. Dist.	
	mean	st dev	mean	st dev	mean	st dev		mean	st dev	mean	st dev	mean	st dev
T1.1	0.740	0.073	0.803	0.000	0.783	0.033	T1.2	0.554	0.038	0.598	0.000	0.567	0.052
	0.564	0.206	0.713	0.100	0.706	0.112		0.553	0.037	0.536	0.052	0.537	0.053
	0.682	0.063	0.614	0.118	0.641	0.147		0.503	0.003	0.506	0.001	0.536	0.053
	0.740	0.073	0.757	0.000	0.633	0.154		0.554	0.038	0.536	0.000	0.522	0.018
	0.647	0.169	0.617	0.148	0.694	0.109		0.538	0.027	0.527	0.025	0.530	0.026
	0.564	0.022	0.591	0.082	0.638	0.118		0.503	0.002	0.502	0.000	0.513	0.020
	0.740	0.073	0.661	0.000	0.570	0.123		0.554	0.038	0.528	0.000	0.522	0.018
	0.633	0.028	0.515	0.127	0.551	0.105		0.515	0.019	0.542	0.011	0.528	0.027
	0.513	0.011	0.553	0.060	0.608	0.092		0.503	0.002	0.502	0.000	0.522	0.019
Max.	0.740	0.206	0.803	0.148	0.783	0.154	Max.	0.554	0.038	0.598	0.052	0.567	0.053
Min.	0.513	0.011	0.515	0.000	0.551	0.033	Min.	0.503	0.002	0.502	0.000	0.513	0.018
Avg.	0.647	0.080	0.647	0.071	0.647	0.110	Avg.	0.531	0.023	0.531	0.010	0.531	0.032
T1.3	0.737	0.078	0.818	0.000	0.775	0.074	T1.4	0.794	0.087	0.882	0.000	0.818	0.110
	0.607	0.182	0.658	0.158	0.659	0.158		0.647	0.204	0.679	0.190	0.679	0.191
	0.627	0.057	0.565	0.108	0.608	0.183		0.630	0.060	0.565	0.109	0.629	0.219
	0.737	0.078	0.731	0.000	0.614	0.117		0.794	0.087	0.794	0.000	0.642	0.145
	0.629	0.117	0.591	0.124	0.654	0.132		0.670	0.121	0.616	0.159	0.698	0.170
	0.502	0.000	0.539	0.065	0.593	0.121		0.502	0.000	0.543	0.072	0.613	0.158
	0.737	0.078	0.663	0.000	0.583	0.078		0.794	0.087	0.708	0.000	0.611	0.084
	0.571	0.084	0.558	0.093	0.556	0.093		0.585	0.109	0.607	0.088	0.589	0.107
	0.502	0.000	0.527	0.043	0.610	0.094		0.502	0.000	0.525	0.040	0.639	0.119
Max.	0.737	0.182	0.818	0.158	0.775	0.183	Max.	0.794	0.204	0.882	0.190	0.818	0.219
Min.	0.502	0.000	0.527	0.000	0.556	0.074	Min.	0.502	0.000	0.525	0.000	0.589	0.084
Avg.	0.628	0.075	0.628	0.066	0.628	0.117	Avg.	0.658	0.084	0.658	0.073	0.658	0.145
T1.5	0.608	0.129	0.756	0.000	0.697	0.096	T1.6	0.607	0.127	0.753	0.000	0.695	0.096
	0.603	0.126	0.611	0.122	0.612	0.129		0.601	0.127	0.609	0.122	0.610	0.128
	0.544	0.037	0.532	0.046	0.591	0.143		0.542	0.037	0.532	0.046	0.590	0.142
	0.608	0.129	0.548	0.000	0.538	0.010		0.607	0.127	0.548	0.000	0.535	0.011
	0.547	0.021	0.533	0.016	0.533	0.025		0.545	0.022	0.531	0.016	0.533	0.025
	0.506	0.003	0.511	0.015	0.522	0.024		0.506	0.003	0.511	0.013	0.521	0.024
	0.608	0.129	0.521	0.000	0.521	0.004		0.607	0.127	0.520	0.000	0.520	0.004
	0.518	0.003	0.524	0.003	0.517	0.011		0.516	0.003	0.522	0.002	0.516	0.010
	0.501	0.000	0.508	0.008	0.515	0.011		0.501	0.000	0.507	0.008	0.514	0.011
Max.	0.608	0.129	0.756	0.122	0.697	0.143	Max.	0.607	0.127	0.753	0.122	0.695	0.142
Min.	0.501	0.000	0.508	0.000	0.515	0.004	Min.	0.501	0.000	0.507	0.000	0.514	0.004
Avg.	0.560	0.064	0.560	0.023	0.561	0.050	Avg.	0.559	0.064	0.559	0.023	0.559	0.050
T1.7	0.599	0.129	0.748	0.000	0.689	0.100							
	0.592	0.133	0.605	0.123	0.605	0.126							
	0.533	0.035	0.528	0.040	0.587	0.139							
	0.599	0.129	0.537	0.000	0.525	0.011							
	0.535	0.024	0.524	0.011	0.526	0.018							
	0.506	0.002	0.507	0.008	0.517	0.018							
	0.599	0.129	0.512	0.000	0.510	0.002							
	0.512	0.002	0.510	0.001	0.509	0.004							
	0.501	0.000	0.508	0.008	0.515	0.011							
Max.	0.599	0.133	0.748	0.123	0.689	0.139							
Min.	0.501	0.000	0.507	0.000	0.509	0.002							
Avg.	0.553	0.065	0.553	0.021	0.554	0.048							

To analyse the effect of randomness in the selection of the training samples from the dataset, analyses T1.1 to T1.7 are repeated 4 times, with the same inner parameters changing the training set. Table 4.7 shows that, on average, three combinations outperform the others, achieving the same value of the performance metrics (the ROC AUC is here reported, but the assessment is confirmed by the accuracy value): minimum level of the detector radius and the censoring distance (both equal to 0.0101) and any level of self radius (combinations 1, 4 and 7, highlighted in Table 4.7).

Table 4.7: Statistics of the ROC AUC considering repetitions with different training sets.

T1.1		T1.2		T1.3		T1.4		T1.5		T1.6		T1.7	
mean	st dev	mean	st dev	mean	st dev	mean	st dev	mean	st dev	mean	st dev	mean	st dev
0.799	0.016	0.614	0.016	0.834	0.011	0.908	0.024	0.809	0.066	0.808	0.068	0.797	0.065
0.798	0.016	0.616	0.018	0.833	0.012	0.906	0.024	0.804	0.068	0.804	0.069	0.794	0.064
0.748	0.007	0.507	0.001	0.696	0.005	0.703	0.010	0.606	0.027	0.606	0.027	0.599	0.027
0.799	0.016	0.614	0.016	0.834	0.011	0.908	0.024	0.809	0.066	0.808	0.068	0.797	0.065
0.736	0.009	0.507	0.001	0.660	0.006	0.656	0.009	0.584	0.018	0.583	0.018	0.579	0.020
0.588	0.006	0.504	0.001	0.504	0.001	0.504	0.001	0.516	0.006	0.515	0.006	0.515	0.005
0.799	0.016	0.614	0.016	0.834	0.011	0.908	0.024	0.809	0.066	0.808	0.068	0.797	0.065
0.610	0.009	0.505	0.001	0.504	0.001	0.504	0.001	0.521	0.006	0.521	0.006	0.519	0.006
0.520	0.007	0.504	0.001	0.503	0.001	0.503	0.001	0.502	0.001	0.502	0.001	0.502	0.001
0.751	0.011	0.543	0.005	0.746	0.015	0.802	0.018	0.555	0.010	0.554	0.009	0.549	0.010
0.468	0.009	0.548	0.013	0.504	0.007	0.503	0.003	0.551	0.014	0.550	0.015	0.539	0.015
0.691	0.014	0.505	0.003	0.607	0.008	0.622	0.009	0.542	0.012	0.540	0.012	0.534	0.014
0.751	0.011	0.543	0.005	0.746	0.015	0.802	0.018	0.555	0.010	0.554	0.009	0.549	0.010
0.750	0.011	0.570	0.014	0.746	0.015	0.809	0.023	0.555	0.011	0.554	0.009	0.547	0.011
0.577	0.011	0.505	0.003	0.503	0.002	0.503	0.001	0.508	0.004	0.507	0.004	0.507	0.002
0.751	0.011	0.543	0.005	0.746	0.015	0.802	0.018	0.555	0.010	0.554	0.009	0.549	0.010
0.645	0.014	0.505	0.003	0.537	0.006	0.539	0.005	0.524	0.008	0.523	0.008	0.520	0.007
0.531	0.011	0.504	0.003	0.503	0.001	0.503	0.001	0.502	0.001	0.502	0.002	0.502	0.002
0.685	0.028	0.531	0.002	0.690	0.019	0.716	0.016	0.522	0.001	0.520	0.003	0.517	0.005
0.440	0.011	0.532	0.003	0.507	0.006	0.532	0.026	0.525	0.003	0.523	0.000	0.516	0.006
0.648	0.034	0.503	0.001	0.573	0.004	0.575	0.007	0.515	0.006	0.514	0.005	0.511	0.004
0.685	0.028	0.531	0.002	0.690	0.019	0.716	0.016	0.522	0.001	0.520	0.003	0.517	0.005
0.465	0.015	0.554	0.005	0.486	0.011	0.529	0.026	0.525	0.005	0.521	0.004	0.515	0.003
0.569	0.034	0.503	0.001	0.503	0.001	0.503	0.001	0.505	0.003	0.505	0.003	0.505	0.003
0.685	0.028	0.531	0.002	0.690	0.019	0.716	0.016	0.522	0.001	0.520	0.003	0.517	0.005
0.685	0.028	0.544	0.009	0.694	0.020	0.731	0.016	0.522	0.002	0.520	0.003	0.516	0.006
0.533	0.037	0.503	0.001	0.503	0.001	0.503	0.001	0.501	0.001	0.501	0.001	0.502	0.001

Over the repetitions, combination 2 sometimes produced the same performance or outperforms the others. As an example, this combo results the best on average for T1.2. This is because combination 2 is extremely close to combination 1, since the two radii have almost the same value, thus the censoring distance is almost the same $0.5(r_{det} + r_{self}) \approx r_{det}$. However, the exact value is slightly lower in combination 2 than in combination 1. In this case, the classifier shows an aggressive behaviour that, continuing with the immune system analogy, corresponds to an autoimmune disease: namely the classifier identifies many self elements as abnormal. This is due to a value of the censoring distance (e.g.

distance between the training self elements and the detectors) lower than the detector radius (distance around the detectors where a nonself element is supposed to be). Thus, during the training, the detectors are moved away from the self elements to a distance equal to the censoring distance, however, during the monitoring, such detectors still match with the same self elements. Due to the adopted design of the experiment, this happens for the combinations 2, 11, 14, 20 and 23. Therefore, it is recommended to avoid such combinations. It is finally worth noting that the standard deviation induced by the change of training set (always less than 0.017) is larger than - or of the same order of - the change due to maturity age and adaptation rate, confirming that such parameters have low influence in the investigated range. Then, the feature sensitivity and the scaling algorithm effect are tested. The performance metric values, for different scaling algorithms over the same dataset and features, are reported in Table 4.8.

Table 4.8: ROC AUC for different scaling algorithms and features.

T1.1	T2.1	T3.1	T1.2	T2.2	T3.2	T1.3	T2.3	T3.3	T1.4	T2.4	T3.4
MMN	SMS	ZS	MMN	SMS	ZS	MMN	SMS	ZS	MMN	SMS	ZS
0.803	0.754	0.790	0.598	0.696	0.681	0.818	0.819	0.835	0.882	0.886	0.924
0.802	0.748	0.774	0.596	0.695	0.683	0.818	0.817	0.833	0.880	0.888	0.923
0.745	0.711	0.746	0.507	0.567	0.513	0.689	0.756	0.751	0.691	0.816	0.798
0.803	0.754	0.790	0.598	0.696	0.681	0.818	0.819	0.835	0.882	0.886	0.924
0.732	0.698	0.741	0.507	0.555	0.500	0.655	0.748	0.739	0.653	0.802	0.774
0.583	0.636	0.617	0.505	0.500	0.500	0.502	0.639	0.560	0.502	0.647	0.558
0.803	0.754	0.790	0.598	0.696	0.681	0.818	0.819	0.835	0.882	0.886	0.924
0.604	0.648	0.646	0.505	0.500	0.500	0.502	0.663	0.595	0.502	0.675	0.592
0.515	0.571	0.553	0.505	0.500	0.500	0.502	0.544	0.500	0.502	0.546	0.500
0.757	0.711	0.747	0.536	0.633	0.598	0.731	0.757	0.742	0.794	0.823	0.777
0.461	0.408	0.482	0.528	0.574	0.568	0.496	0.583	0.602	0.505	0.657	0.632
0.682	0.689	0.715	0.502	0.532	0.502	0.615	0.711	0.676	0.626	0.746	0.693
0.757	0.711	0.747	0.536	0.633	0.598	0.731	0.757	0.742	0.794	0.823	0.777
0.756	0.711	0.744	0.552	0.638	0.598	0.731	0.771	0.741	0.798	0.823	0.777
0.568	0.623	0.592	0.502	0.500	0.500	0.502	0.600	0.527	0.502	0.604	0.529
0.757	0.711	0.747	0.536	0.633	0.598	0.731	0.757	0.742	0.794	0.823	0.777
0.634	0.672	0.689	0.502	0.506	0.500	0.546	0.672	0.649	0.545	0.703	0.659
0.522	0.564	0.531	0.502	0.500	0.500	0.502	0.526	0.500	0.502	0.528	0.500
0.661	0.694	0.709	0.528	0.575	0.562	0.663	0.690	0.657	0.708	0.739	0.676
0.430	0.418	0.454	0.535	0.574	0.565	0.509	0.583	0.578	0.554	0.600	0.595
0.619	0.670	0.656	0.502	0.521	0.502	0.577	0.655	0.612	0.572	0.671	0.621
0.661	0.694	0.709	0.528	0.575	0.562	0.663	0.690	0.657	0.708	0.739	0.676
0.452	0.428	0.472	0.555	0.596	0.582	0.501	0.576	0.601	0.558	0.600	0.633
0.540	0.628	0.588	0.502	0.500	0.500	0.502	0.595	0.518	0.501	0.599	0.516
0.661	0.694	0.709	0.527	0.575	0.562	0.662	0.690	0.657	0.707	0.739	0.676
0.661	0.681	0.707	0.536	0.590	0.582	0.664	0.703	0.657	0.707	0.746	0.676
0.501	0.583	0.526	0.500	0.500	0.500	0.500	0.519	0.500	0.500	0.523	0.500

Irrespective of the different factors, the best combinations are again 1, 4 and 7 (Table 4.8). The comparison demonstrates that for the same combination of parameters and the same analysed features, the ZS tends to outperform the MMN, whereas the performance of the SMS is scattered, being in some

cases the best, in others the worst of the three. It is worth mentioning that the MMN provides the best overall performance in the comparison among T1.1-T2.1-T3.1, whereas, in the other comparisons, the ZS has always the best overall performance.

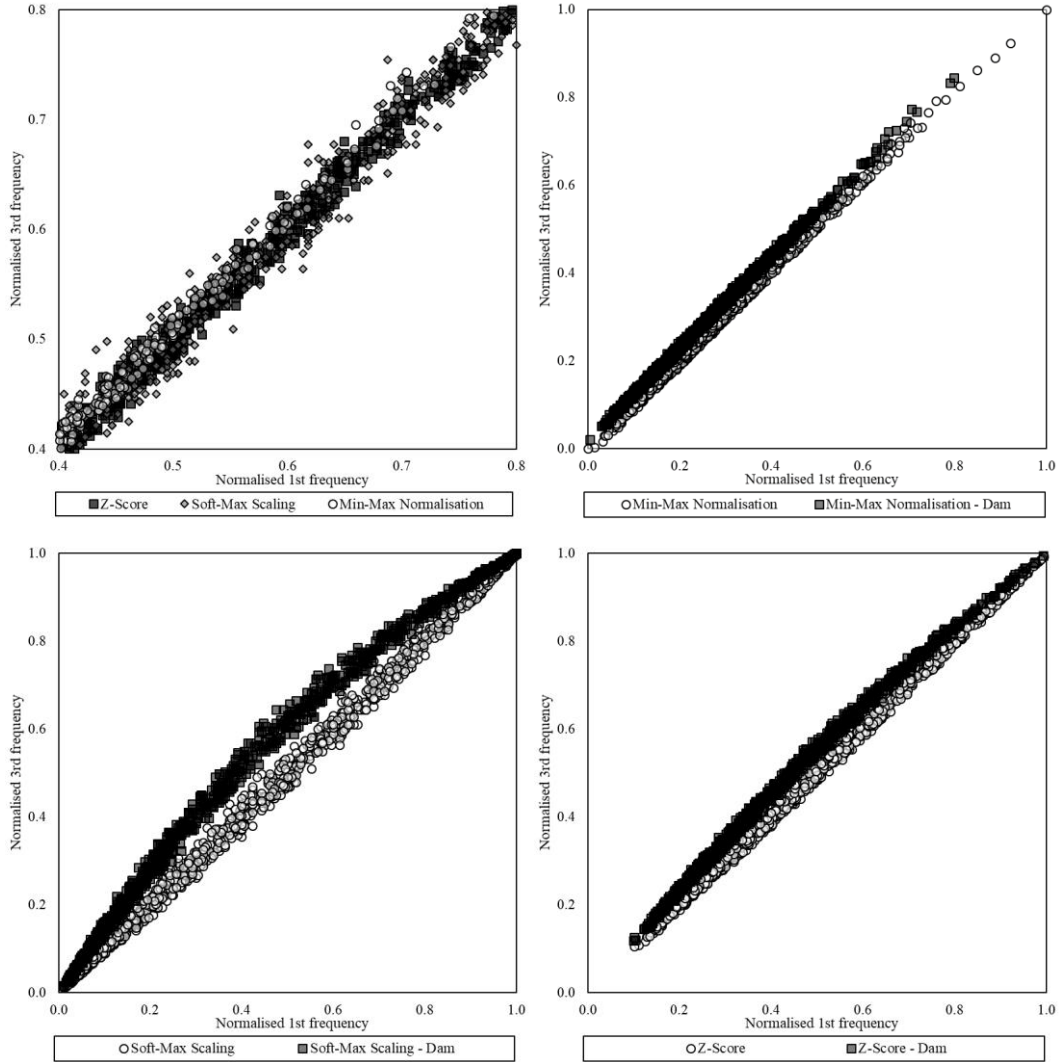


Figure 4.2: Comparison among normalisation algorithms, dataset based on the numerical model of a three-story building, full dataset.

Such results can be explained by considering the operating mode of the three algorithms. The MMN algorithm normalises the features values without introducing any distortion in the shape, whereas the other two algorithms magnify the distance between elements belonging to two different states, especially close to the mean feature values. The features belonging to the damaged states are normalised considering the maximum and minimum values, the mean and the standard deviation, per each feature, calculated based only on the self samples. Comparing the undamaged samples with one of the damaged set (Figure 4.2), SMS algorithm provides the highest scatter of the samples, but also the ZS algorithm

scatter is larger than the MMN. The larger scatter is balanced by a magnification of the distance between the undamaged and damaged datasets, although this magnification decreases closer to the extreme self samples. A similar trend, although less pronounced, appears also in the Z-Score results. The MMN, instead, does not deform the distance between the sets. The same samples are used to test how the decimation and the change in the sample set statistics affect the normalisation performance. The results in Figure 4.3 and Figure 4.4 analyse two decimated dataset (from 1005 to 105 samples). The values for the statistics of the artificially generated natural frequencies before normalisation, in the three cases, are reported in Table 4.9. In the different cases, only a slight shift in the reciprocal position of the samples is likely to happen, due to the change in the mean values.

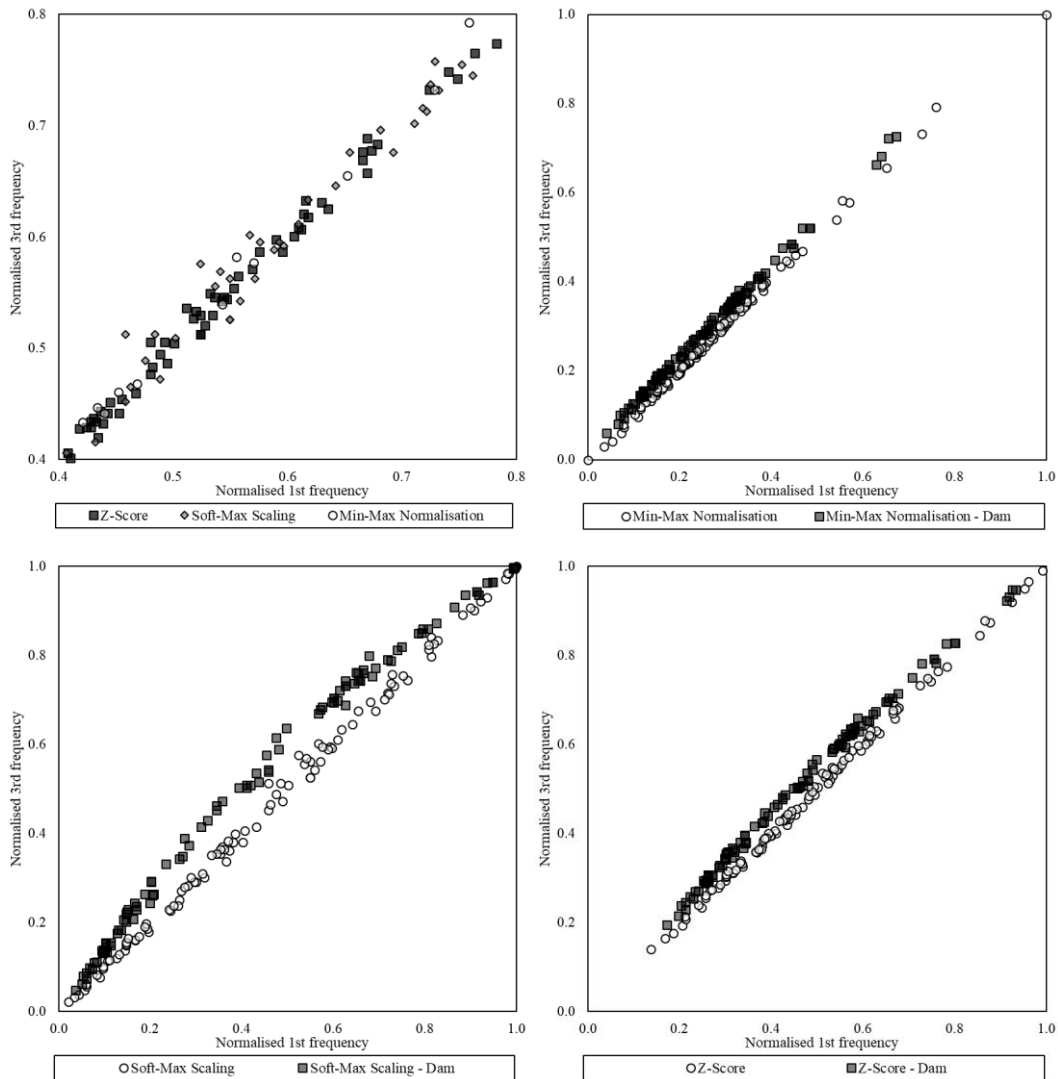


Figure 4.3: Comparison among normalisation algorithms, dataset based on the numerical model of a three-story building, 1st decimated dataset.

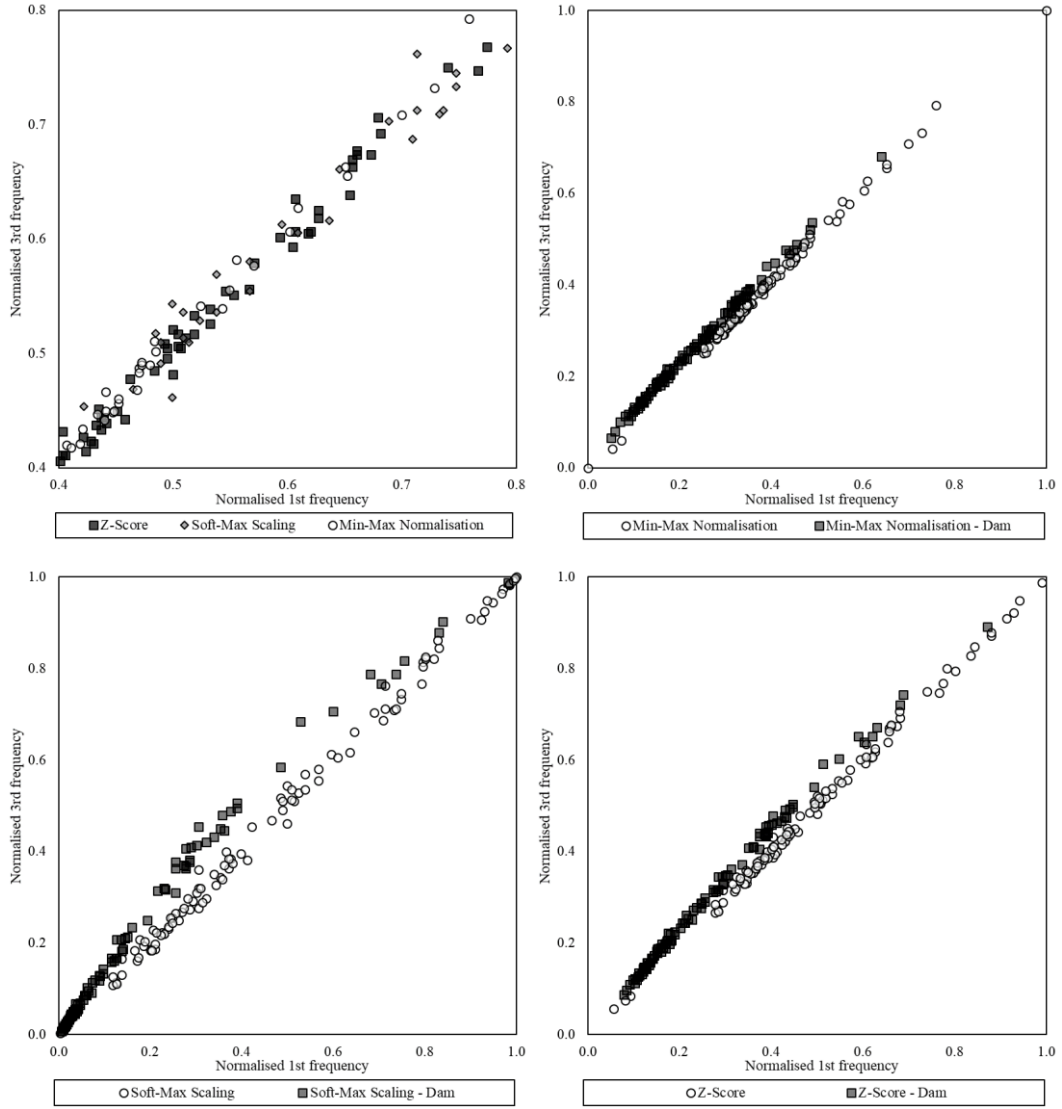


Figure 4.4: Comparison among normalisation algorithms, dataset based on the numerical model of a three-story building, 2nd decimated dataset.

Table 4.9: Statistics for the three different datasets.

Dataset:	Full dataset	1 st Decimated dataset	2 nd Decimated dataset
		Undamaged 1 st natural frequency	
Max:	6.395	6.395	6.395
Min:	5.599	5.599	5.599
Mean:	5.836	5.819	5.903
St.Dev:	0.109	0.120	0.108
		Undamaged 3 rd natural frequency	
Max:	9.909	9.909	9.909
Min:	8.897	8.897	8.897
Mean:	9.203	9.179	9.291
St.Dev:	0.143	0.156	0.140

The algorithm performance, estimated according to the area under the ROC curve, does not take into account the distribution of the samples. Indeed, this index is based on the true positive and the true negative ratios. However, the natural frequency value populations do not present a uniform distribution.

Thus, for the same value of the performance metrics (namely the same value of the true positive and the true negative rate), one classifier might outperform the others because the samples misclassified are less likely to happen. For the sake of simplicity, hereafter, this hypothesis is analysed considering only the case of damage D1 in the 1st-3rd frequency feature space. The artificially generated populations of these two frequencies and their distributions are shown in Figure 4.5. For the following analysis, two log-normal distributions are fitted to the samples.

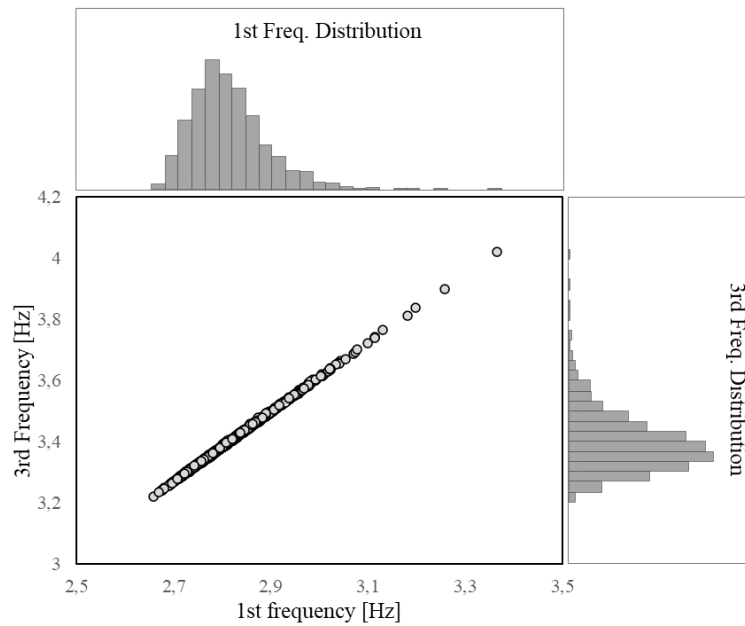
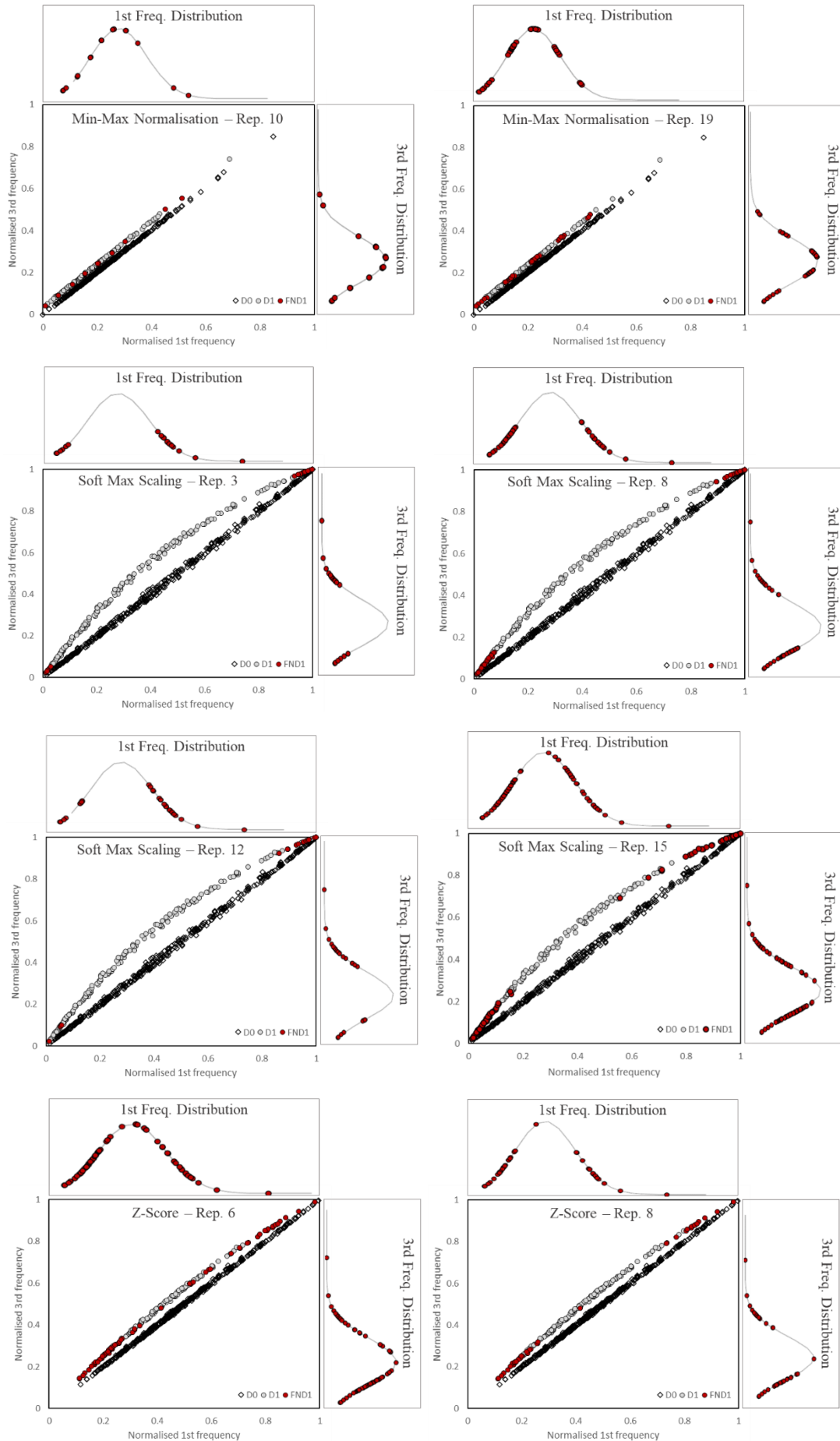


Figure 4.5: Distribution of the 1st and 3rd natural frequencies with their representation in the 2-D space before normalisation.

Table 4.10: Confusion matrix and performance metrics for different scaling algorithms and instances.

CASE	TN	FP	TP-D1	FN-D1	TPR	TNR	AUC
Min-Max Normalisation							
10	197	4	180	21	0.896	0.980	0.938
19	200	1	135	66	0.672	0.995	0.833
Soft Max Scaling							
3	200	1	180	21	0.896	0.995	0.945
8	201	0	146	55	0.726	1.000	0.863
12	201	0	168	33	0.836	1.000	0.918
15	201	0	114	87	0.567	1.000	0.784
Z-score							
6	201	0	119	82	0.592	1.000	0.796
8	201	0	155	46	0.771	1.000	0.886
12	201	0	176	25	0.876	1.000	0.938
17	201	0	167	34	0.831	1.000	0.915

For the three scaling methodologies, some instances are selected (Table 4.10). These are not the most sensitive to damage D1 as they present a very high true negative rate (TNR) in most cases equal to 1, but a lower true positive rate (TPR). The TPR is lowered by a varying number of false negative errors, between 20 and 90 out of the 201 used to validate the algorithm instance.



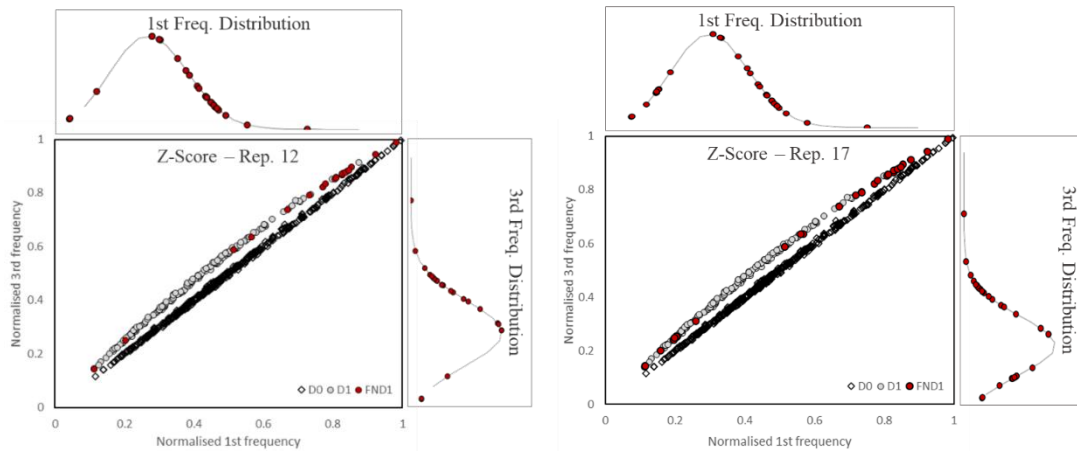


Figure 4.6: Normalised feature space with D0 and D1 samples for different algorithm instances and log-normal distributions for the two features. In red misclassified damaged samples.

Figure 4.6 reports the 2-D plot of the normalised feature space, with the misclassified elements highlighted in red both in the plot and in the graph of the log-normal distribution. It is worth noting that the elements misclassified by the NSA after the Soft Max Scaling are gathered in the tails of the distribution, namely their probability of occurrence is lower whereas, in the region where most of the samples fall, there are no mistakes. For the Z-score this phenomenon is less evident and several misclassified samples belong to the centre of the distribution.

Focusing on a single instance per each scaling algorithm, namely the repetition 10 for Min-Max Normalisation, repetition 3 for Soft Scaling and repetition 12 for Z-Score, it is possible to further analyse this difference. In these three instances, the number of false negatives is 21, 21 and 25 respectively.

Figure 4.7 shows the probability density function for the 1st and the 3rd natural frequencies. The false negatives are highlighted in red and the area of the distribution where they fall is indicated in grey. For repetition 10 of the Min-Max Normalisation, this area encompasses the mode of the distribution, and corresponds to almost the whole area under the curve. For repetition 12 of Z-Score, the mode is almost reached on the right. For repetition 3 of Soft Max Scaling, finally, the false negatives all fall in an area with lower probability.

In the example, the Soft Max Scaling has less false negatives. Nonetheless, the misclassified samples continue to fall close to the tails of the distribution even for more false negatives, as in the case of the repetition 8 of Soft Max Scaling and Z-Score, with 55 and 46 false negatives, respectively. In this example, as shown in Figure 4.8, even though the number of false negatives is larger for the Soft Max Scaling, the probability of misclassification is lower than for the Z-Score.

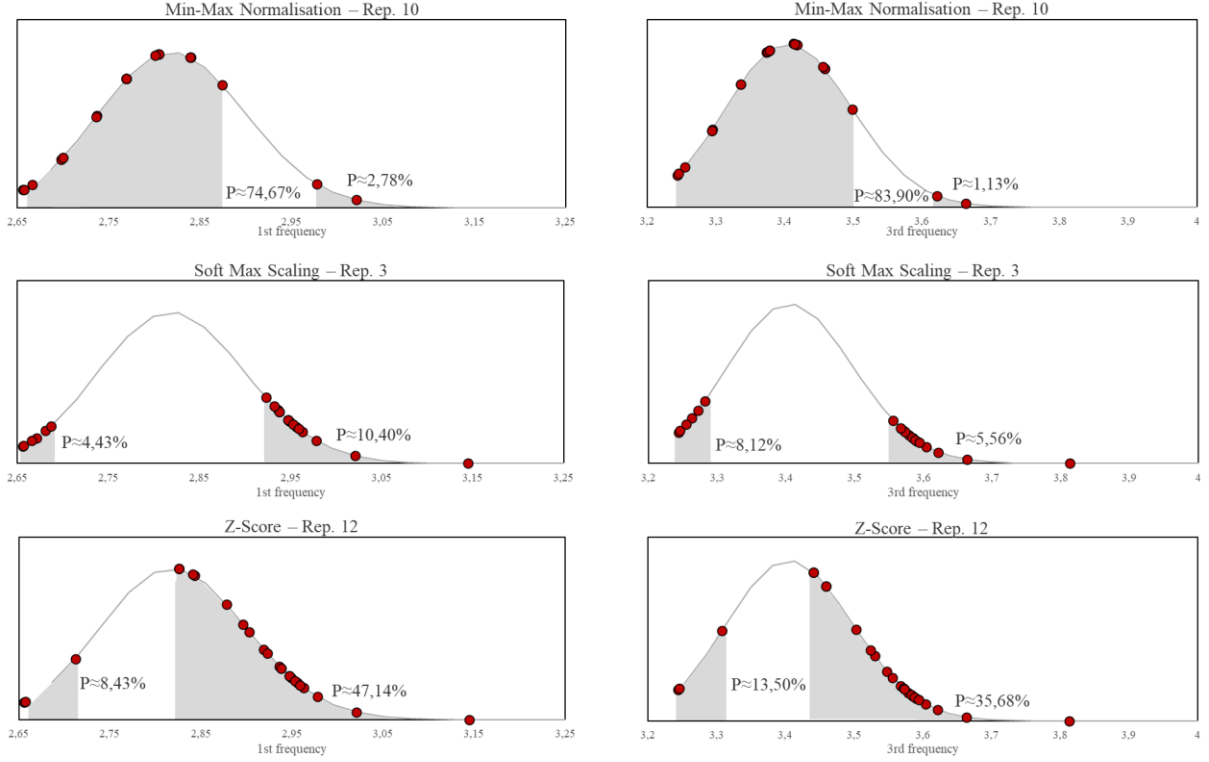


Figure 4.7: log-normal probability density function of the 1st and 3rd natural frequencies. In red the false negatives and in grey the probability of the area of the distribution where the false negatives fall. First example: repetition 10 of Min-Max normalisation, repetition 3 of Soft-Max scaling and repetition 12 of Z-score.

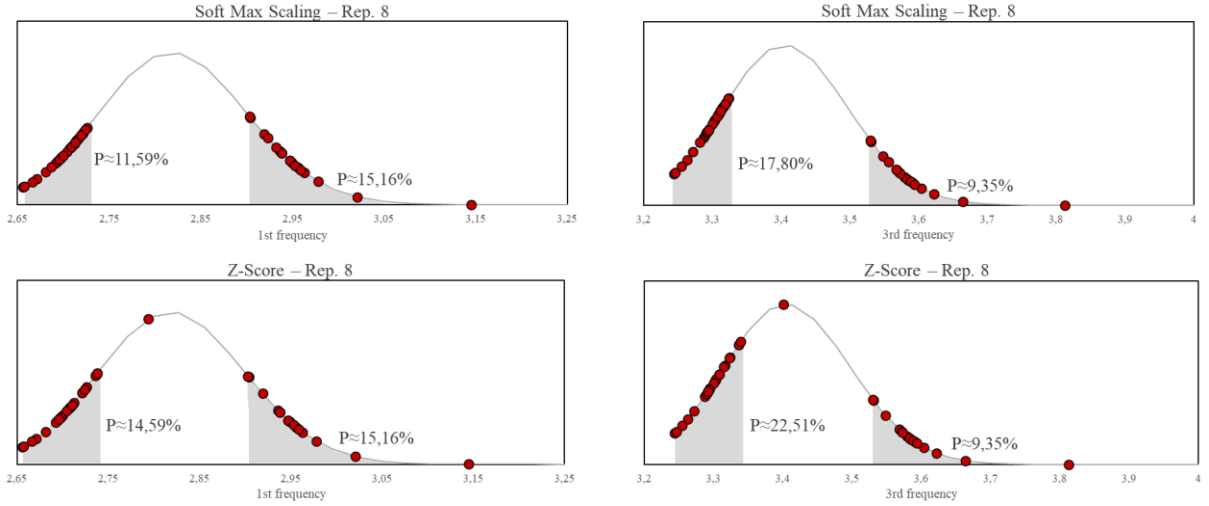


Figure 4.8: log-normal probability density function of the 1st and 3rd natural frequencies. In red the false negatives and in grey the probability of the area of the distribution where the false negatives fall. Second example: repetition 8 of Soft-Max scaling and Z-score.

It is worth noting that the original distribution of the frequencies is created artificially and does not resemble a real one. Moreover, the formulation of both an alternative performance metric and/or an alternative scaling algorithm is out of the scope of the present work. Nonetheless, the apparent advantage of the Soft Max Scaling technique and, in general, a magnification of the distance between samples in

the most likely area deserve more research, considering real world feature distributions and require a new performance index that takes the distribution of the samples into account.

Finally, the performed analyses allow to investigate the damage sensitivity of different QoI. For simplicity, only the datasets after Min-Max Normalisation are discussed, but similar conclusion can be generalised to the other scaling algorithms. According to the overall performance reported in Table 4.7 and Table 4.8, the feature spaces, ordered from the most to the least effective pairs, are: (1) 2nd natural frequency – 3rd natural frequency; (2) 1st natural frequency – 3rd natural frequency; (3) temperature – 1st natural frequency; (4) temperature – 2nd natural frequency; (5) delta frequencies; (6) temperature – 3rd natural frequency; (7) 1st natural frequency – 2nd natural frequency.

In the present case study, it was noted that whenever a single natural frequency is mapped against the temperature, scenarios are often barely distinguishable. An interesting correlation between the natural frequencies under varying environmental conditions with a very low scatter emerges. Such result suggests that the plot of two frequencies, one against the other, allows to define a reliable model of the system behaviour under varying external factors, even when a direct monitoring of such factors is not carried out. However, a pair of frequencies does not always guarantee a proper sensitivity. As reported above, for instance, the classifiers in the feature space formed by the 1st and the 2nd frequencies have the lowest overall performance. Indeed, the 1st and 2nd natural frequencies belong to the first translational modes in the two directions. Being the lowest global modes, they are likely to be less sensitive to small local damages and this might justify the lower performance.

Nonetheless, the promising results achieved by using pairs of frequencies suggests that including more frequencies could increase the information content, thus the performance of the classification. However, due to the characteristics of the applied DGNSA, the feature space cannot exceed the two dimensions. A viable solution consists in the use of the frequency distance. In this feature space, damages that induce uniform frequency shift for all the investigated modes are likely undetectable, whereas the others are likely to be more evident. In the present case study, three artificially generated frequencies are observed, thus three possible features based on frequency difference can be considered: (1) 2nd minus 1st; (2) 3rd minus 1st; (3) 3rd minus 2nd.

Due to the characteristics of the system, the first two frequency values are very close and the more symmetric the system, the closer they are; thus, their difference is not deemed a reliable feature. This value, in fact, is very close to zero, but once scaled to the range [0,1], small errors are likely to be magnified (ill-conditioned problem). Therefore, the feature formed by their difference is not employed in the present study.

The above results analyse the overall performance, namely the confusion matrix is calculated including all the samples from damage scenarios in a single damaged class. Nonetheless, it is not possible to exclude that distinct features have different sensitivity to the scenarios. This different sensitivity to the damage can be used to improve the detection methodology, on one side to provide an estimation of the likelihood of the damage whereas, on the other side, to reach a higher order classification. Unlike the previous analyses, in this case, the methodology can only be used under supervised learning, namely it is assumed an *a-priori* knowledge of the structural dynamic response in one or more damage scenarios. This approach has been called substructuring in chapter 3 and applies more classifiers in parallel, each one optimised and tailored to a specific feature space. In the following, four classifiers, namely four substructures of the global classifier, perform in parallel in the following feature spaces:

- delta frequencies;
- 1st natural frequency – 2nd natural frequency;
- 1st natural frequency – 3rd natural frequency;
- 2nd natural frequency – 3rd natural frequency.

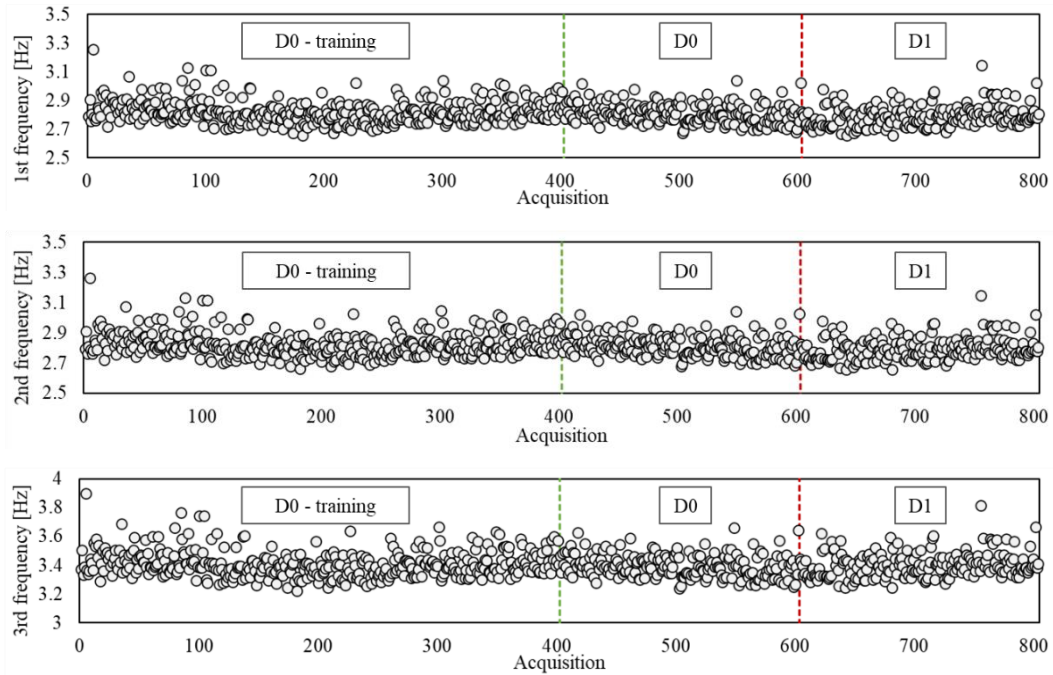


Figure 4.9: Training data followed by the validating data, undamaged and damaged in scenario 1 respectively.

The 4 classifiers are trained over 402 samples, where each one is a vector composed of the 1st, 2nd and 3rd frequency at a given time step, considering more algorithm parameter values. 201 undamaged and 201 damaged samples per each damage scenario are used for validation, in order to define the best classifier in each feature space based on a full factorial design of the experiment. Figure 4.9 shows, for

instance, the data used for training and validation in the case of scenario D1. Indeed, this first stage corresponds to the first repetition of instances T1.1 to T1.4. Table 4.11 reports the confusion matrix of the classifiers that are selected after the validation. In this table the classifier is identified also through the feature space and the number of the combination case of the full factorial design. The performance metrics over the validation data that are significant for the following analysis are reported in Table 4.12. It is worth comparing the overall performance, as showed in Table 4.8, with the results of Table 4.12. The classifier in the delta frequencies feature space (instance T1.1) does not provide a large overall performance (see Table 4.8), due to a low performance against scenarios D1, D2 and D3 (namely damages to the internal pillar), balanced by the best performance against scenarios D4, D5 and D6 (i.e. damages to the external pillar). Similarly, it is worth noting that the classifier in the feature space composed by the 1st and the 2nd frequency has the lowest overall performance, but it is particularly sensitive to scenario D5. As expected, the total number of FNs increases with the scenarios and it is particularly high for D3, D4, D5, this is due to the typology of damage. Indeed, D1 affecting a central pillar at the first story is likely to influence more the global response of the structure than damages at the upper floors or on a lateral pillar.

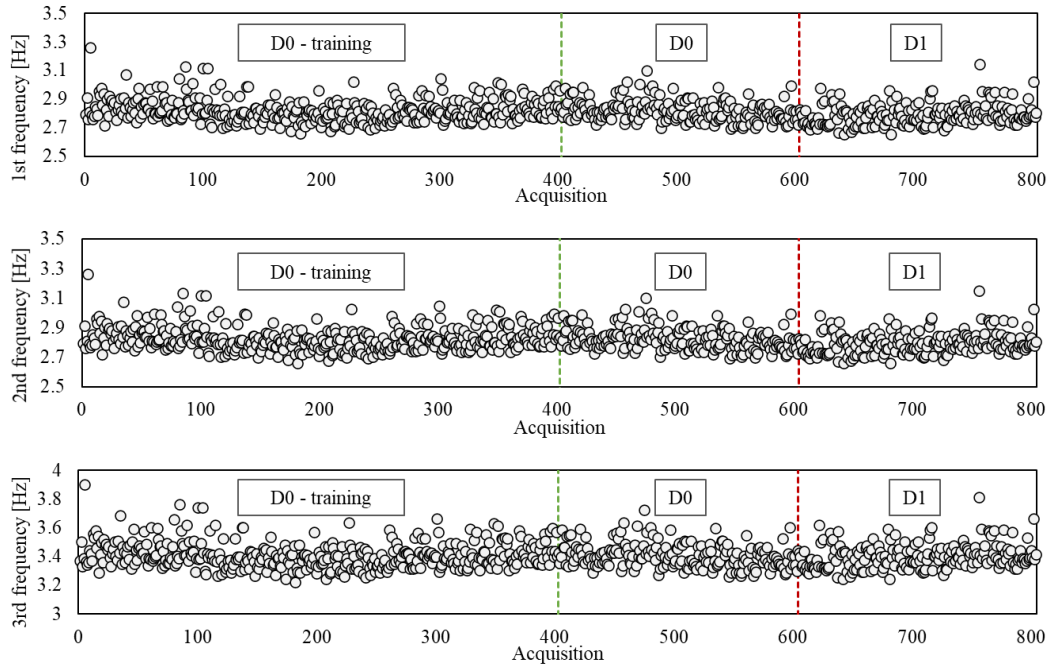
Table 4.11: Confusion matrix for the best classifier in each feature space.

Feature space	Comb.	TN	FP	TP1	FN1	TP2	FN2	TP3	FN3	TP4	FN4	TP5	FN5	TP6	FN6
delta frequencies	1	187	14	77	124	55	146	44	157	201	0	201	0	201	0
1st-2nd freq.	1	188	13	9	192	11	190	16	185	42	159	145	56	39	162
1st-3rd freq.	2	192	9	201	0	201	0	199	2	128	73	23	178	38	163
2nd-3rd freq.	1	195	6	201	0	201	0	199	2	176	25	88	113	68	133

The four classifiers are then applied in parallel to the analysis of a test dataset composed of 201 undamaged and 201 damaged samples per each scenario. For instance, Figure 4.10 shows the data used for damage scenario D1. In order to reproduce a scenario similar to a real-world, continuous monitoring the temperature gradient is followed along a continuous cycle from 10°C to 30°C and then back to 10°C. Hence, the data belonging to D0 vary in the range from 10 to 30°C, followed by the data belonging to D1 varying between 30° to 10°C. The performance of the detection system is tested by identifying after how many samples the classifiers fire an alarm, the reliability of the alarm and the capability of identifying what kind of damage emerged.

Table 4.12: Performance metrics for the 4 classifiers in the different damage scenarios. For each scenario, the feature space with the best performance is highlighted.

Scenario	Feature space	Comb.	TPR	TNR	AUC	Ranking
D1	delta frequencies	1	0.383	0.930	0.657	3
	1st-2nd freq.	1	0.045	0.935	0.490	4
	1st-3rd freq.	2	1.000	0.955	0.978	2
	2nd-3rd freq.	1	1.000	0.970	0.985	1
D2	delta frequencies	1	0.274	0.930	0.602	3
	1st-2nd freq.	1	0.055	0.935	0.495	4
	1st-3rd freq.	2	1.000	0.955	0.978	2
	2nd-3rd freq.	1	1.000	0.970	0.985	1
D3	delta frequencies	1	0.219	0.930	0.575	3
	1st-2nd freq.	1	0.080	0.935	0.507	4
	1st-3rd freq.	2	0.990	0.955	0.973	2
	2nd-3rd freq.	1	0.990	0.970	0.980	1
D4	delta frequencies	1	1.000	0.930	0.965	1
	1st-2nd freq.	1	0.209	0.935	0.572	4
	1st-3rd freq.	2	0.637	0.955	0.796	3
	2nd-3rd freq.	1	0.876	0.970	0.923	2
D5	delta frequencies	1	1.000	0.930	0.965	1
	1st-2nd freq.	1	0.721	0.935	0.828	2
	1st-3rd freq.	2	0.114	0.955	0.535	4
	2nd-3rd freq.	1	0.438	0.970	0.704	3
D6	delta frequencies	1	1.000	0.930	0.965	1
	1st-2nd freq.	1	0.194	0.935	0.565	4
	1st-3rd freq.	2	0.189	0.955	0.572	3
	2nd-3rd freq.	1	0.338	0.970	0.654	2

**Figure 4.10: Training data followed by the test data, undamaged and damaged in scenario 1 respectively.**

Hereafter, for convenience, it is assumed that the classification score of each of the classifier in parallel is 0 when the sample is labelled self and 1 when it is labelled nonself. Several ways of fusing the single classifications into a global score exist, being this a current research topic (e.g. [275], [276]). An in depth investigation on this field falls outside of this thesis scope. Therefore, two very simple approaches are tested to demonstrate the possible advantages of the multi-classifiers combination, leaving any further improvement as recommended future developments.

The first approach consists in an averaged sum of the classification score of the four classifiers. The global classification score CS is the following:

$$CS = (\sum_{i=1}^4 CS_i) / 4 \quad (\text{Eq. 4.37})$$

As mentioned above, the classifier performs differently against distinct damage scenarios, thus the second approach aims at rewarding the classifiers which are expected to perform better, penalising the others. This is achieved through a weighted sum:

$$CS = (\sum_{i=1}^4 CS_i \cdot AUC_i) / \sum_{i=1}^4 AUC_i \quad (\text{Eq. 4.38})$$

where AUC_i is the ROC AUC of the i -th classifier for the same damage scenario as calculated during the validation.

The trend of the two global classification score formulations over the test data progression in the six damage scenarios is shown in Figure 4.11. Irrespective of the scenario, the visual inspection of the graphs allows to identify a clear variation of the trend after the damage outbreak. Such variation is minimum for the scenarios D5 and D6 that induce a smaller change in the structural behaviour. Yet, it is possible to set a threshold to automatise the detection. It is worth mentioning that the global classifier has a good performance even when the simple classifiers has a very low one. For instance, the classifier in the space given by 1st and 2nd natural frequency has a very low TPR for most of the scenarios (e.g. for D1 is 0.045), thus it classifies most of the damaged samples as healthy. Nonetheless, the sample is likely to be correctly labelled by the global classifier since the low performance of one classifier is balanced by the others. Between the two formulations, the weighted sum provides larger classification scores for the true positive samples. In some extreme cases it provides larger scores also for the false positive, but on average the distance between the classification score of the samples before and after the damage outbreak is increased by the weighted sum approach which is, thus, considered more reliable. Nonetheless, it is worth noting that the implementation of the weighted sum as formulated above requires that the actual damage scenario is known during the monitoring, since the classification score depends on the specific ROC AUC. Moreover, the previous knowledge of the same damage scenario during the validation stage is

required in order to calculate the ROC AUC. The simple averaged sum has the advantage that this larger knowledge is not needed. Yet, it requires that nonself data are available for the validation, in order to select the best classifier in each feature space.

The observations so far allow to answer the first two research questions, namely after how many samples the classifiers fire an alarm and the reliability of the alarm itself. Indeed, in all the analysed scenarios, a shift is evident in the classification score trend ever since the damage outbreak. Before the damage outbreak, the maximum number of consecutive samples with a classification score higher than 0 is 5 whereas, after the damage outbreak, all the samples have a higher score, except for 4 samples in scenario D3, only two of them consecutive. The importance of the consecutive classification score is analysed in more detail in chapter 5 through the use of proper counters to reduce the risk of false alarm, as explained in chapter 3.

The last research question that is addressed in the present section examines the capability of identifying what kind of damage emerged, namely the capability of performing a higher order classification. The same problem is faced also by relying on the signatures as damage sensitive features and more details are provided in section 4.3.3. However, in this section, it is solved by employing the distinct sensitivity of the classifiers in parallel. The same test data are processed and classified in the four feature spaces and moving True Negative and True Positive Rates are calculated. The moving TNR and TPR over n latest samples are calculated assuming that the last n samples are actual negatives and actual positives, respectively. For instance, if $n = 3$ and the i -th classifier produces, between the instant $t - 2$ and t , the classification score $\{1,0,0\}$, viz. positive, negative, negative, the two metrics are $TNR_{i,n}(t) = 0.66$ and $TPR_{i,n}(t) = 0.33$, respectively. This approach does not require any assumptions on the actual state; indeed, two options are tested for any new sample, namely that the last n samples belong, in one case, to the healthy state and, in the other, to the damaged state. As these metrics consider n previous samples, they are expected to fluctuate around a constant value when the system is in a steady state and to vary after the damage outbreak. This transient state is due to the fact that the previous samples do not belong all to the same system condition. The identification approach is based on the comparison between the moving TNR and TPR and the TNR and TPR calculated during the validation in the different damage scenarios.

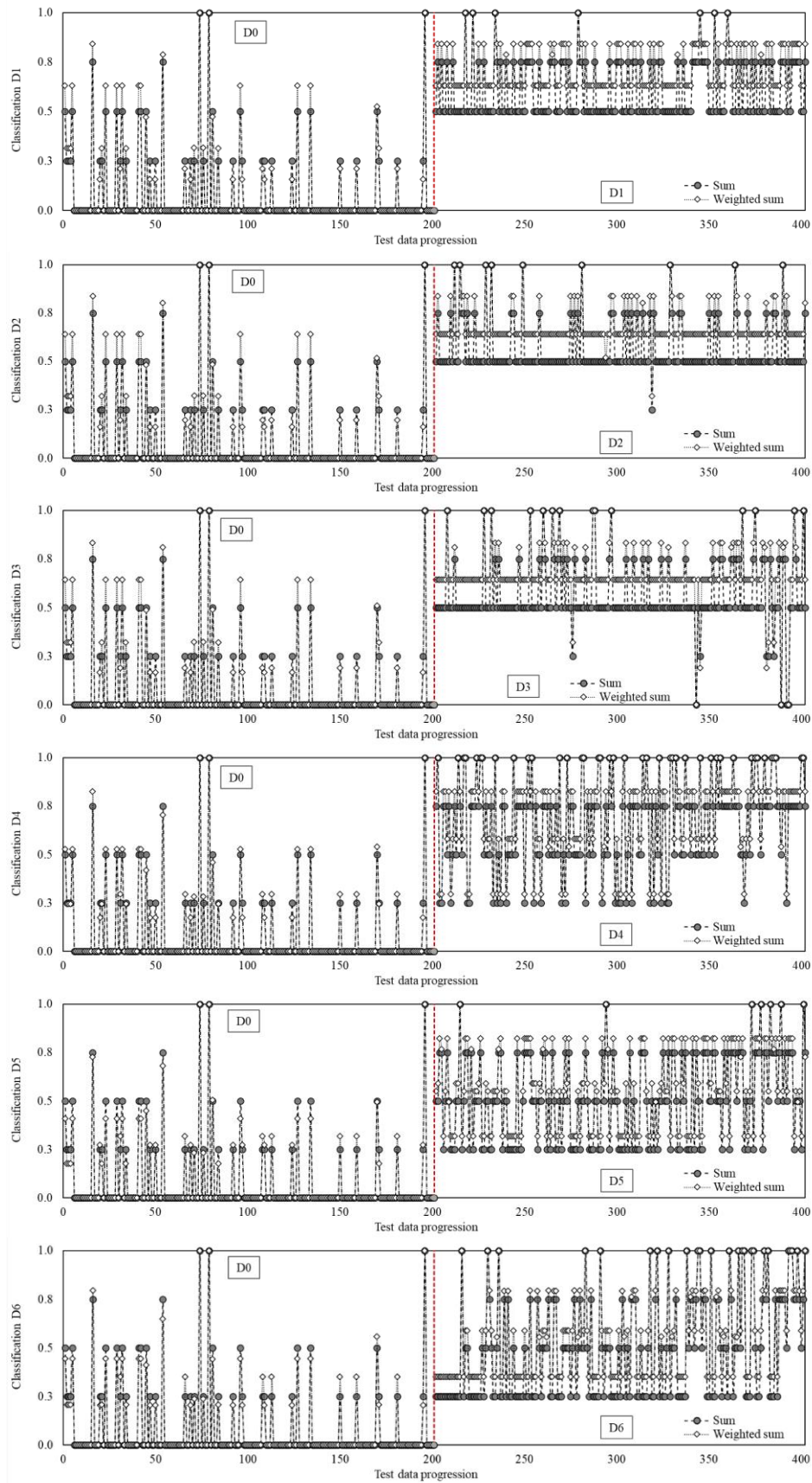


Figure 4.11: Trend of the global classification scores based on averaged sum and weighted sum over the test data.

Two different values of n are tested, namely 12 and 24. Independently of the damage scenario, the TNR of the validation stage is constant, thus it can be used as indicator of the damage outbreak. For instance, considering the case of damage scenario D1, the trend of $TNR_{i,12}(t)$ is shown in Figure 4.12a and the trend of $TNR_{i,24}(t)$ in Figure 4.12b. For the sake of clarity, in these figures, the values referring to the classifier in the delta frequencies space is called c1, in the 1st-2nd frequencies space is called c2, in the 1st-3rd frequencies space is called c3 and in the 2nd-3rd frequencies space is called c4. Moreover, the index 'v' in the label of the curves refers to the constant value achieved during the validation stage and here used as reference. It is evident, after the damage outbreak, the transient state in the value for all the classifiers that diverge from the corresponding TNR of the validation, excluding the classifier c2 that is the least sensitive. The transient is evident for all the damage scenarios, as for instance in D6 that induces a lower change in the structural behaviour (see Figure 4.13). The main difference between the case of $n = 12$ and $n = 24$, as expected, is that an increasing number of samples reduces the variance in the fluctuation during the steady state although it increases the length of the transient period.

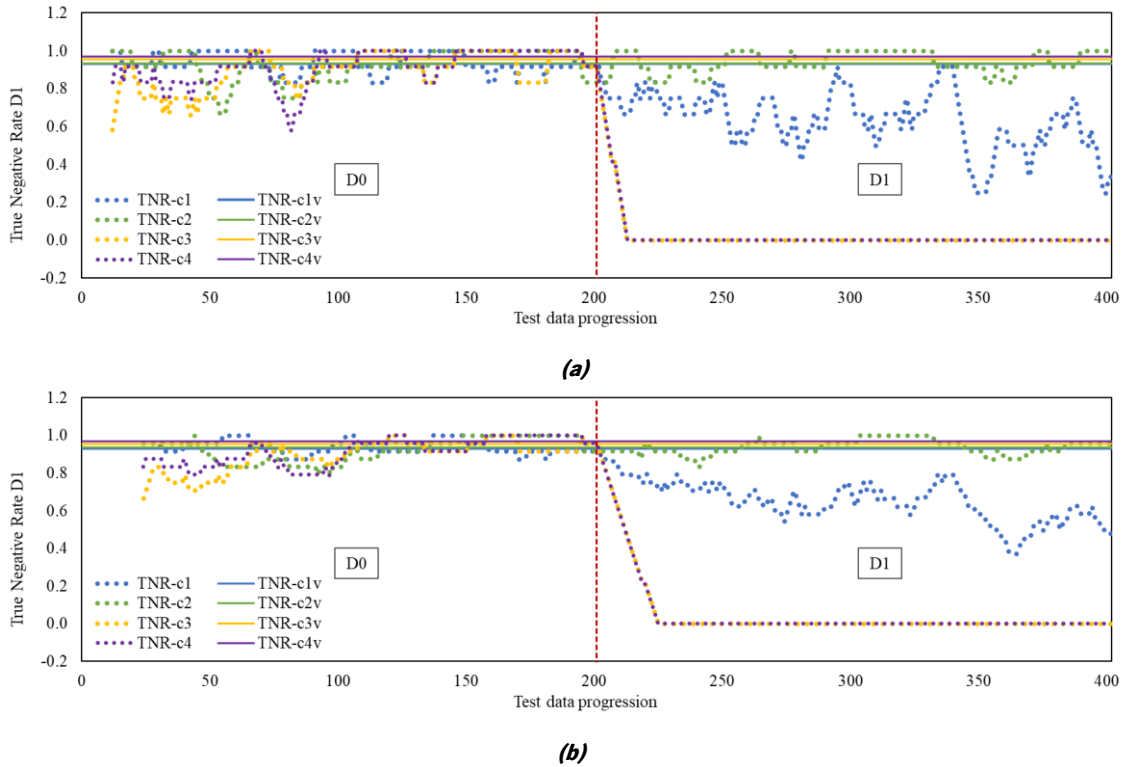


Figure 4.12: Trend of the moving true negative rate over the test data in scenario D1: (a) average over 12 samples; (b) average over 24 samples.

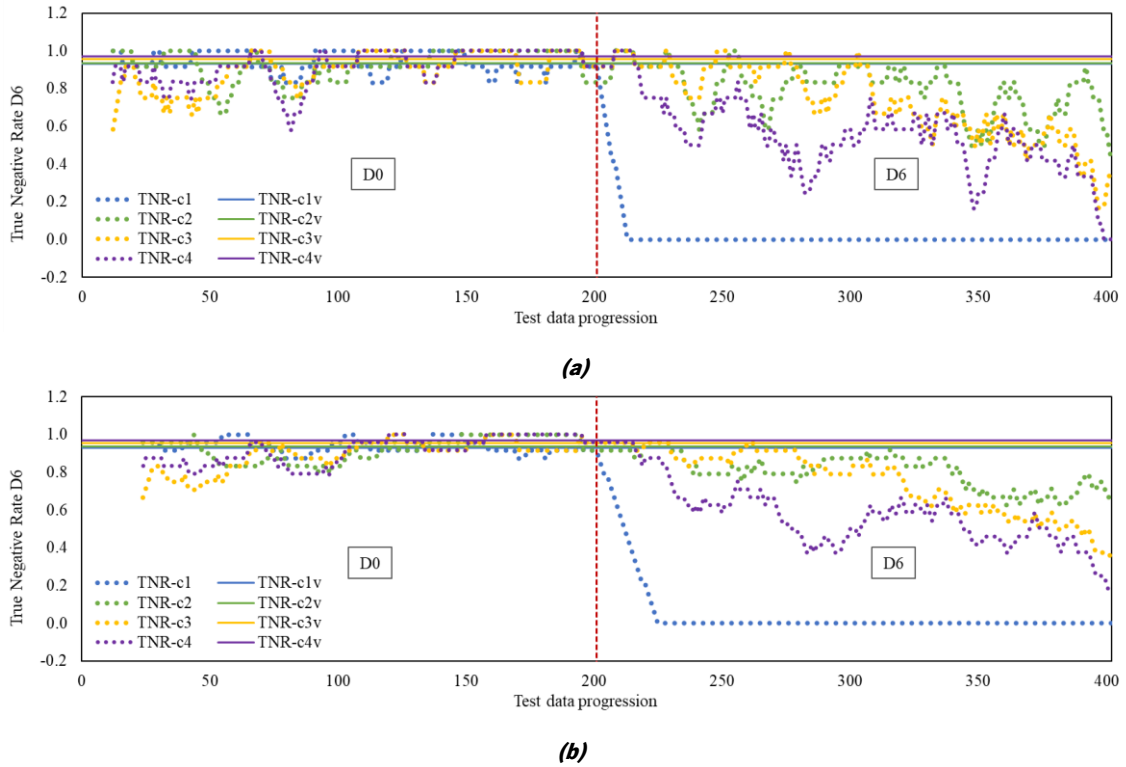


Figure 4.13: Trend of the moving true negative rate over the test data in scenario D6: (a) average over 12 samples; (b) average over 24 samples.

Then, the $TPR_{i,12}(t)$ and $TPR_{i,24}(t)$, for each damage scenario, are compared with the corresponding values calculated during the validation. For instance, Figure 4.14 presents the result for scenario D1 of $TPR_{i,24}(t)$. The moving TPR of each one of the classifier is compared with the constant TPR calculated for the same classifier during validation (index 'v' in the label of the curves). In the case of the TNR a single scenario exists (viz. the undamaged), whereas, in the case of the TPR, the moving metrics calculated during the real-time monitoring must be compared with all the scenarios that are available during the validation. As expected, as long as the system is healthy, the moving TPR is negligible, as most of the classifiers in parallel label the samples as negative. After the outbreak of damage D1 (damage in the central pillar at first floor), there is an evident transition after which the moving TPR is expected to fluctuate around an average value very close to the TPR of the actual damage scenario calculated during the validation. The visual inspection of the graphs allows to qualitatively confirm that the trend of the four $TPR_{i,24}(t)$ presents a good correspondence with the TPR achieved over the validating dataset in scenario D1 (Figure 4.14a). Indeed, the moving TPR-c2 remains very low, being this classifier poorly sensitive to the scenario D1. The moving TPR-c1 shows an average sensitivity, whereas TPR-c3 and TPR-c4, being more sensitive, increase significantly with negligible fluctuation. A good agreement is still visible in Figure 4.14b, namely with the TPR achieved over validating dataset D2, likely due to a similar change induced by the two damages (D2 is a damage in the central pillar at second floor). The large fluctuation

of the moving TPR-c1 is the main reason for the possible misidentification. The other TPRs (Figure 4.14c-f) present increasing differences. This provides a qualitative support for the higher order classification. In the example of Figure 4.14, one might conclude that the structure is damaged and, based on the comparison with the known TPRs from different scenarios, that the damage is probably located in the central pillar, either at the second floor or, most likely, at the first one. This approach requires a supervised learning, hence, an *a-priori* knowledge of the behaviour induced by different damage scenarios is needed. The validity of the approach is further confirmed by the comparison between the performance matrix in each damage scenario and each classifier, in the validation and in the test, as reported in Table 4.13. Here, the performance metrics related to the test datasets are not the moving values but the overall, considering all the 201 undamaged and 201 damaged samples. A good agreement is visible between the results during validation and during test, in terms of TNR, TPR and overall AUC, for the same feature space and damage scenario.

Table 4.13: Comparison of the performance metrics per each damage scenario and classifier during the validation and during the test.

Scenario	Feature space	Comb.	Validation			Test		
			TPR	TNR	AUC	TPR	TNR	AUC
D1	delta frequencies	1	0.383	0.930	0.657	0.378	0.940	0.659
	1st-2nd freq.	1	0.045	0.935	0.490	0.045	0.925	0.485
	1st-3rd freq.	2	1.000	0.955	0.978	1.000	0.881	0.940
	2nd-3rd freq.	1	1.000	0.970	0.985	1.000	0.910	0.955
D2	delta frequencies	1	0.274	0.930	0.602	0.204	0.940	0.572
	1st-2nd freq.	1	0.055	0.935	0.495	0.060	0.925	0.493
	1st-3rd freq.	2	1.000	0.955	0.978	0.995	0.881	0.938
	2nd-3rd freq.	1	1.000	0.970	0.985	0.995	0.910	0.953
D3	delta frequencies	1	0.219	0.930	0.575	0.209	0.940	0.575
	1st-2nd freq.	1	0.080	0.935	0.507	0.114	0.925	0.520
	1st-3rd freq.	2	0.990	0.955	0.973	0.970	0.881	0.925
	2nd-3rd freq.	1	0.990	0.970	0.980	0.955	0.910	0.933
D4	delta frequencies	1	1.000	0.930	0.965	1.000	0.940	0.863
	1st-2nd freq.	1	0.209	0.935	0.572	0.204	0.925	0.565
	1st-3rd freq.	2	0.637	0.955	0.796	0.642	0.881	0.761
	2nd-3rd freq.	1	0.876	0.970	0.923	0.836	0.910	0.873
D5	delta frequencies	1	1.000	0.930	0.965	1.000	0.940	0.955
	1st-2nd freq.	1	0.721	0.935	0.828	0.458	0.925	0.692
	1st-3rd freq.	2	0.114	0.955	0.535	0.090	0.881	0.485
	2nd-3rd freq.	1	0.438	0.970	0.704	0.502	0.910	0.706
D6	delta frequencies	1	1.000	0.930	0.965	1.000	0.940	0.955
	1st-2nd freq.	1	0.194	0.935	0.565	0.214	0.925	0.570
	1st-3rd freq.	2	0.189	0.955	0.572	0.279	0.881	0.580
	2nd-3rd freq.	1	0.338	0.970	0.654	0.473	0.910	0.692

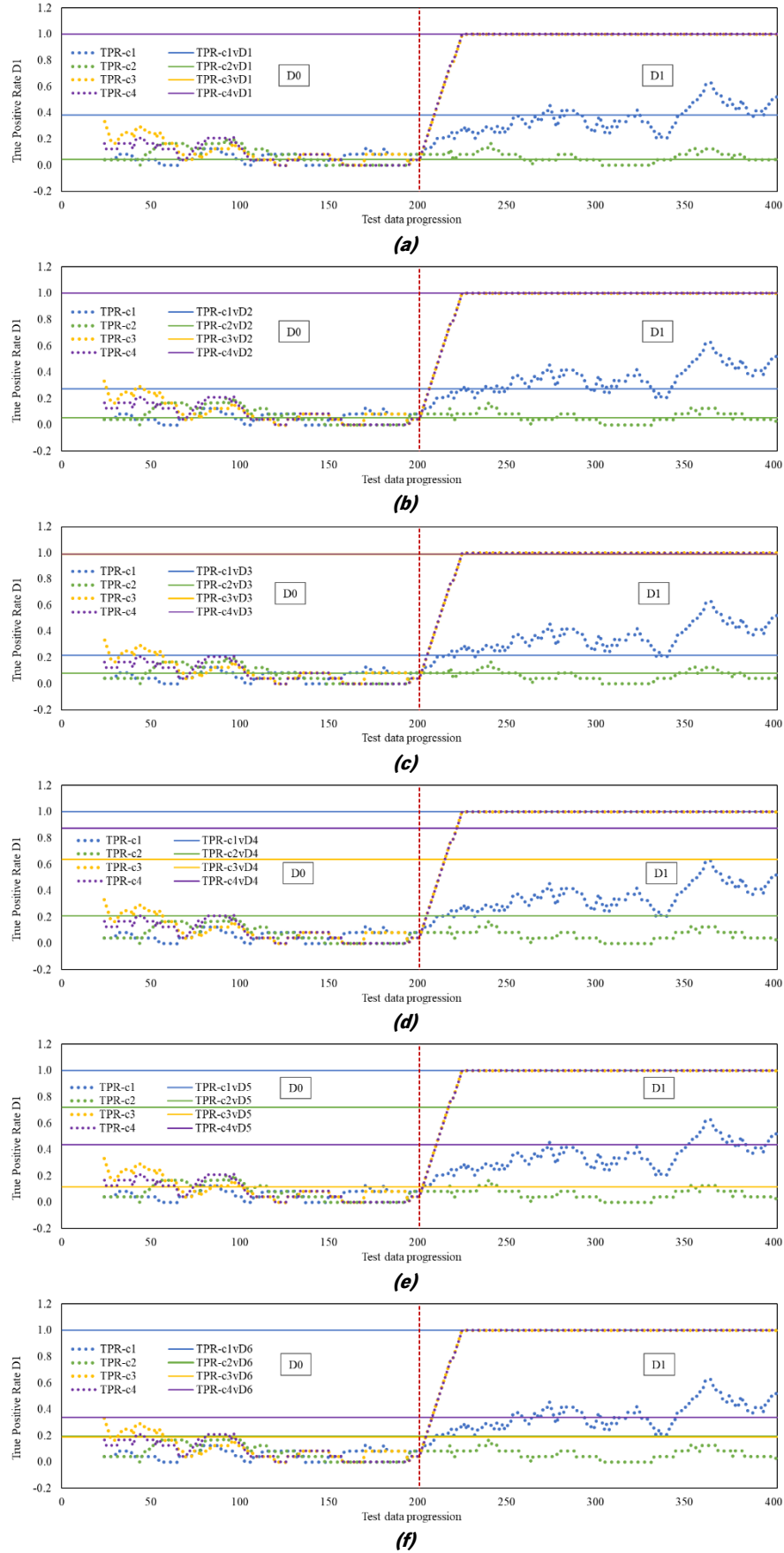


Figure 4.14: Trend of the moving true positive rate over the test data in scenario D1, compared with: (a) TPR over validating dataset D1; (b) TPR over validating dataset D2; (c) TPR over validating dataset D3; (d) TPR over validating dataset D4; (e) TPR over validating dataset D5; (f) TPR over validating dataset D6.

Concluding, it is worth noting that the approach here presented is rather qualitative, although a quantitative formulation of the comparison between moving $TPR_{i,n}(t)$ and TPR during validation might be easily implemented. At this stage of the work, the promising results are highlighted but further applications to real-world case studies and a comparison with well-known high order classification methods are both necessary and recommended as future scope.

4.3. I-40 bridge over Rio Grande, New Mexico (USA)

4.3.1. Description of the case study

The present case study consists in a set of artificial data generated through a FE model of the well-known I-40 bridge over the Rio Grande, in New Mexico (USA). In 1993, this structure was dynamically tested under ambient and forced vibrations both in its actual conditions and inducing intentionally increasing extent of damage before the final demolition [277]. The experimental data set has been analysed in the past by many researchers using different damage detection methods becoming a benchmark in the SHM field of research [278].

The I-40 bridge consisted of two separated twin spans, one for each traffic direction, divided into three independent sections with three spans each, two shorter at the ends, with a length of 39.9 m, and one larger in the middle, with a length of 49.7 m. The structural section of the bridge presents a concrete deck supported by three steel stringers, set over floor beams and two steel girders.

The FE model that is used to generate the data was developed in the software ANSYS, at the Technical University of Madrid [279], based on [280], and reproduces the 3 spans of the bridge which carried the east bound traffic (Figure 4.15).

To acquire pseudo-experimental data in agreement with [277], four damage scenarios, D1 to D4, are numerically generated to replicate the damage introduced in the real structure by gradually cutting the middle span of the north plate girder as follows:

- (1) D1: crack simulated in the web, a plate element joins the two halves to shorten the crack, the flange is intact;
- (2) D2: long crack simulated in the web, the flange is intact;
- (3) D3: crack also in the flange, a beam element half the width of the flange connects either side of the crack;
- (4) D4: the flange and web are severed completely.

For each damage scenario, the real field test is simulated by introducing an excitation to the location of the shaker and by recording the response in 26 points coincident with the location of the accelerometers. The simulated force excitation has a constant spectrum amplitude of 10000 N/Hz^2 , within a frequency range of $[2.2, 6.0] \text{ Hz}$. All data were sampled at 33.3 Hz (six natural frequencies of the bridge are below 5 Hz) for a total duration of 60 s , resulting in 2000 data points per channel. The generated pseudo-experimental data set is composed of a single record per sensor and per damage scenario. Then, each feature is copied and polluted with a normal distributed random noise R according to the following equation:

$$r^*(f) = r(f) + r(f) \cdot R(f) \cdot \text{err} \quad (\text{Eq. 4.39})$$

where $r^*(f)$ is the polluted amplitude of the signature for the frequency line f , $r(f)$ is the available value of the feature for the same frequency f , $R(f)$ is a number randomly generated from a distribution with zero mean and standard deviation $\sigma = 1$ and err is the noise level. For each spectral line, a different number $R(f)$ is used. The training set is composed of 100 self elements (dynamic signatures) whereas the validating set is composed of 100 different self elements and 100 nonself elements per each damage scenario.

The main results of the testing campaign that is discussed hereafter have been already published by the author [281].

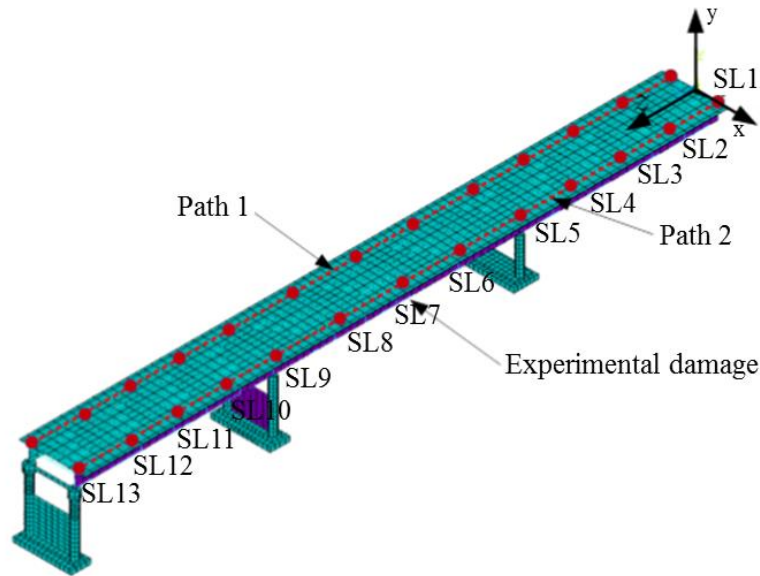


Figure 4.15: Portion of the bridge under investigation, with the location of the damage and the points used to extract the dynamic response.

4.3.2. Testing campaign

The aim of the experimental campaign is to test the following factors:

- Two performance metrics: area under the Precision-Recall curve and area under the ROC curve;
- Three sensor locations, SL , on different spans and at different distances from the damage: SL7 is the closest, being located at the central midspan, SL10 is at intermediate distance, on the third span, and SL2 is the farthest, on the first span;
- Five levels of noise, err : 1%, 5%, 10%, 15% and 20%;
- Two damage-sensitive features: the power spectral density (PSD) and the absolute value of the fast Fourier Transform (absFFT) of the output acceleration(s), both in the frequency domain;
- Three frequency ranges of interest, Δf : [2-3] Hz (including two natural frequencies), [2.2-2.4] Hz (around the first natural frequency) and [2.2-4.5] Hz (including five natural frequencies).

After this preliminary analysis, the higher order classification based on the anomaly score, as explained in chapter 3, is investigated.

The frequency range is investigated as it affects the time requirement. The input is a force with a constant spectrum in the range 2-6 Hz, namely the range of the lowest modes of the structure, thus none of the output ranges analysed can overcome such bounds. Compared to the first range, the second and the third ranges are respectively shorter and larger, thus the time required, for the analysis in each of the cases can be compared and used as a metric to assess the computational burden of the DGNSA. Also, in the second and third ranges, the lowest frequency line is removed. Since it emerged that this is essential for the detection of the first damage scenario, the performance is expected to change. Excluding the tests performed to assess the influence of the sensor location, all the remaining instances focus on the least sensitive location, i.e. SL2, to demonstrate that a proper damage detection can be carried out even monitoring a point far away from the damage. For the sake of clarity, each problem instance is identified by a specific code as reported in Table 4.14.

Table 4.14: Problem instance code for the I-40 bridge case study.

Description of the parameter	PSD	absFFT
SL = 2, err = 1%, $\Delta f = [2 - 3]$ Hz	T1.1	T2.1
SL = 2, err = 5%, $\Delta f = [2 - 3]$ Hz	T1.2	T2.2
SL = 2, err = 10%, $\Delta f = [2 - 3]$ Hz	T1.3	T2.3
SL = 2, err = 15%, $\Delta f = [2 - 3]$ Hz	T1.4	T2.4
SL = 2, err = 20%, $\Delta f = [2 - 3]$ Hz	T1.5	T2.5
SL = 2, err = 10%, $\Delta f = [2.2 - 2.4]$ Hz	T1.6	–
SL = 2, err = 10%, $\Delta f = [2.2 - 4.5]$ Hz	T1.7	–
SL = 7, err = 10%, $\Delta f = [2 - 3]$ Hz	T1.8	–
SL = 10, err = 10%, $\Delta f = [2 - 3]$ Hz	T1.9	–

A single data scaling approach is used, namely the Min-Max Normalisation with $\pm 20\%$ of acceptable range for the signature amplitude. This procedure sometimes cuts off some elements belonging to the second and third damage scenarios as their amplitude falls out of the range, thus the nonself elements vary between 391 and 400 for the different problem instances (only testing the smaller frequency range the positive elements decrease to 374), whereas the number of self elements is always 200. The algorithm instance is always optimised by a three-level full factorial design which considers the following parameters:

- Censoring distance, $dist$;
- Detector radius, r_{det} ;
- Self radius, r_{self} .

The effect of randomness in the process is assessed by performing 2 repetitions per dataset.

4.3.3. Results

In the present section, the main results of the performed analyses are discussed. In Figure 4.16, the plots of the performance metrics corresponding to the instance T1.3 are provided as examples.

Regarding the two performance metrics investigated, the PRAUC does not seem suitable for assessing the performance as it rewards aggressive classifiers. Indeed, it increases for a reduction in FNs but ignores the increasing number of FPs.

As expected, the performance decreases with the distance of the sensor from the damage. Moreover, its variance increases. The closest location (SL7) produces the highest performance and the best combination emerges more clearly. On the contrary, as the distance between sensor and damage increases, more combinations do provide the same performance measure, being impossible to identify a best overall. However, the performance deterioration is not so significant to consider the farthest location analysed (SL2) unsuitable for the damage detection.

The performance decreases also as the noise increases, due to the overlapping of the polluted signatures. By increasing the noise, the algorithm tends to misclassify not only the first damage scenario, but also the second and third. Also, increasing the noise flattens the performance of the different combinations, reducing the variance of the measures and providing the same value of the ROCAUC for more combinations.

Changing the noise level extent, the best parameters' combination changes significantly when the signature is the PSD, whereas, using the absFFT, the best combinations are almost always the same. Thus, the absFFT results more suitable for the damage detection via NSA.

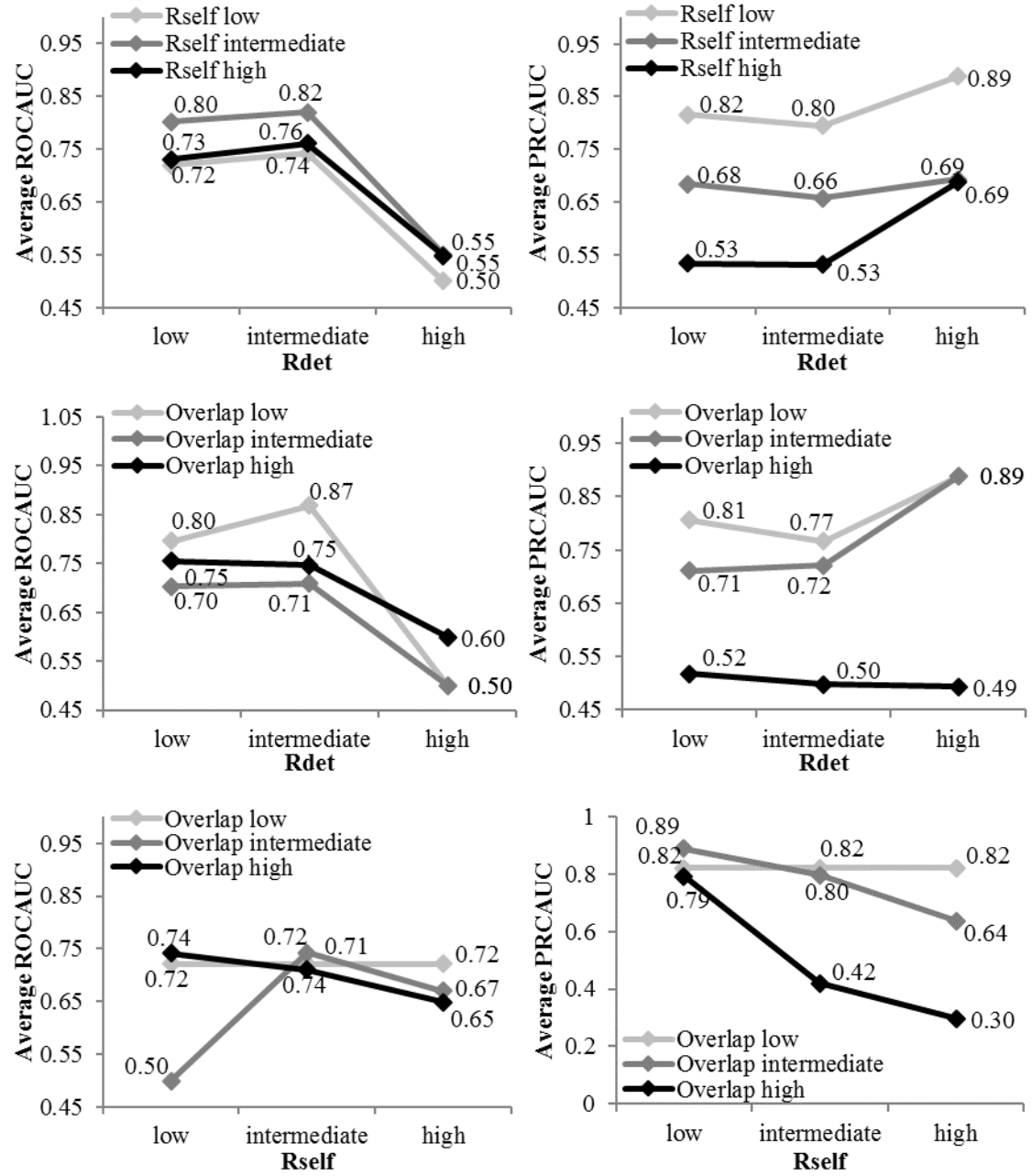


Figure 4.16: Marginal mean plots for the two repetitions of T1.3 (using PSD as signature).

The smallest frequency range provides the highest performance for the best combination, whereas the largest range provides the lowest. The matching charts show that the signatures differ more in the lower frequency range, see for instance the matching charts reported in Figure 4.17. Thus, a larger frequency range does not improve the detection rate but increases remarkably the computational burden, as the number of analysed points per each signature are respectively 12, 64 and 141: the average time consumption for a single combination increases from 22.5 to 105.5 and finally to 221 seconds. The elapsed time for the process also includes the requirement for “overhead” operations, such as generating and censoring the detectors, but does not consider the requirement for plotting all the graphs and charts

of the results. All the tests are run on a server with an Intel Core i9-7900X X-series 3.30 GHz processor and 128 GB of memory run. During the analysis, the CPU usage is limited to 5%. Concluding this preliminary analysis, some of the best classifiers are tested over a larger data set composed of 1000 signatures per each scenario (D0, D1, D2, D3 and D4). For the sake of clarity, the instances are renamed according to the code reported in Table 4.15, together with the final confusion matrix. Excluding the instance I04, here tested to assess the suitability of the anomaly score for multiclass classification, the other instances misclassify less than 5% of the self elements, whereas the percentage of misdetected nonself elements is equal to 25%. Almost all of the nonself signatures wrongly classified belong to the scenario D1. Excluding this scenario, the percentage of misdetection decreases to 0.1% in the worst case.

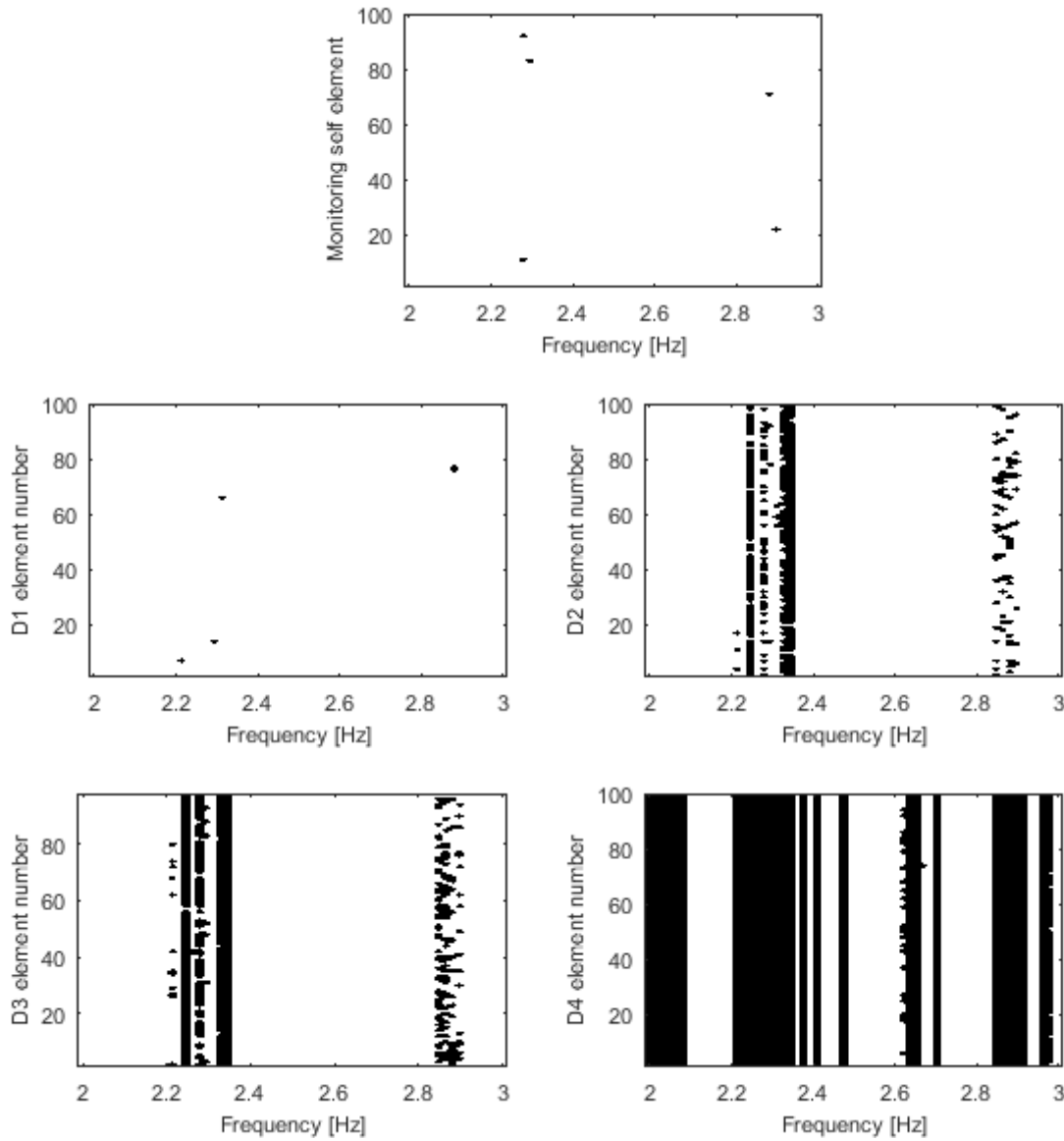


Figure 4.17: Matching chart for the second repetition of T1.3, combination 13, using PSD as signature. In the chart, the points which trigger at least one detector are reported in black.

Table 4.15: Algorithm instances tested over a larger dataset and confusion matrix.

CODE	INSTANCE	COMBINATION	TP	FP	FN	TN	FN1/FN2/FN3/FN4
I01	T1.3	COMB. 13	3047	51	948	1049	948/0/0/0
I02	T1.6	COMB. 17	2905	11	995	1089	994/1/0/0
I03	T2.1	COMB. 1	3947	53	0	1047	0/0/0/0
I04	T2.3	COMB. 1	3993	1100	0	0	0/0/0/0
I05	T2.3	COMB. 3	3017	35	976	1065	972/4/0/0

The instance I03 provides the best result, classifying correctly all the nonself elements and mistaking only 5% of the self elements. It is stressed again that, in this case, the noise level is only 1% . The equivalent combination, but with 10% of noise (I04), classifies all the self elements as anomalous.

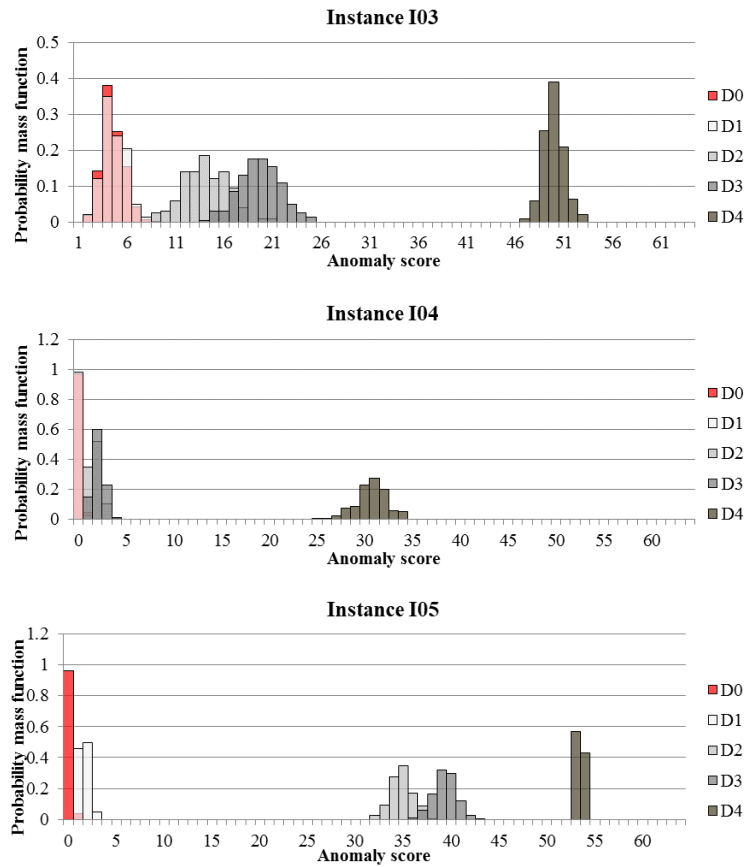


Figure 4.18: Histogram of the anomaly score for the algorithm instances tested for higher order classification. The distribution of the anomaly score is based on the validation dataset considering the 2 repetitions for each instance.

Finally, the classification and quantification skills of the algorithm are tested, by allocating the anomaly score of each monitored element to a specific damage scenario based on the probability mass functions defined in the validation stage. This method is tested using the instances I03, I04 and I05 (see the instances in Table 4.15). As described in chapter 3, the use of the anomaly score for higher order classification is based on the probability mass function of each scenario. It is worth reminding that the

anomaly score is the number of points of the signature matched by at least one detector. The distributions of such values are determined during the validation stage, thus, based on a set of undamaged and damaged samples known before the actual monitoring stage. The histograms of the anomaly scores are reported in Figure 4.18.

Hereafter, the distributions are approximated to Gaussian ones. To carry out the higher order classification, each new element recorded by the acquisition system is compared to the known distributions. Namely, given an element x , and a damage scenario i , its Z score corresponds to the standardized statistic:

$$Z_i = \frac{x - \mu_i}{\sigma_i} \quad (\text{Eq. 4.40})$$

where μ_i and σ_i are, respectively, the mean and the standard deviation of the scenario.

The two-tailed p-value of a Z-test statistic is assumed as an approximation of the probability that the element belongs to the given scenario. Actually, the Z-test, given a sample whose mean is μ_0 , checks the null-hypothesis that this value is equal to the mean of a given normal distribution, $H_0: \mu_0 = \mu_i$

Whenever Z is such that the p-value $P(Z < -Z_{\alpha/2} \text{ or } Z > Z_{\alpha/2}) = \alpha$, for a given significance level α , the null-hypothesis is rejected. However, in this case, a single value rather than a sample set is compared to the known distribution.

Table 4.16: Classification results for I03.

	D0	D1	D2	D3	D4
Mean p-value ∈ D0	0.7979	≈ 0.0	≈ 0.0	≈ 0.0	≈ 0.0
Mean p-value ∈ D1	0.0205	0.4365	≈ 0.0	≈ 0.0	≈ 0.0
Mean p-value ∈ D2	≈ 0.0	≈ 0.0	0.4832	0.0077	≈ 0.0
Mean p-value ∈ D3	≈ 0.0	≈ 0.0	0.0131	0.4911	≈ 0.0
Mean p-value ∈ D4	≈ 0.0	≈ 0.0	≈ 0.0	≈ 0.0	0.3293
p-value < p-value ∈ D0	–	0%	0%	0%	0%
p-value < p-value ∈ D1	4.8%	–	0%	0%	0%
p-value < p-value ∈ D2	0%	0%	–	7.1%	0%
p-value < p-value ∈ D3	0%	0%	1.8%	–	0%
p-value < p-value ∈ D4	0%	0%	0%	0%	–

As an example, Table 4.16 shows the results of the instance I03, which outperforms the others. Indeed, this instance assigns the wrong label to 4.8% of the self elements D0 classified as D1, 1.8% of the D2

elements classified as D3, and 7.1% of the D3 elements classified as D2. Regarding the self elements, when the anomaly score is 1, the probability of belonging to the class D1 is so high (45% against 4% of belonging to healthy state) that the algorithm mislabels them. The errors in labelling the elements of D2 and D3 happen where the distributions overlap (in the present case, D2 anomaly score equals 38 and 39, and D3 anomaly score equals 36 and 37). The same conclusions remain valid for the instances I04 and I05. The former has the same factor combination of I03, but the signatures are polluted with 10% of noise. This induces a higher percentage of errors, particularly relevant for the classes D0 and D1 (almost 67% of the elements of D0 are classified as D1 and more than 35% of the D1 elements are classified as D0), but still acceptable for the larger damages. I04 monitors exactly the same dataset of I03. As already mentioned, I04 is a better binary classifier than I03 which, being too aggressive, considers all the self elements as anomalous. But, in terms of detachment of the anomaly score distribution, I03 becomes more suitable than I04 for multiclass classification. The effect of the overlapping distributions, in fact, strongly affects the performance of I04 that misclassifies almost 97% of D0 elements and 85% of D2 elements.

4.4. Conclusions

In the present chapter, the damage detection methodology was validated by analysing its most relevant features over two artificial datasets. The validation, indeed, is of interest not only for the new developed version of the Negative Selection Algorithm, but also for the other main components, as for instance, the data scaling procedure, the performance metrics and the design of the experiments methodology.

Numerical case studies allowed to simulate damage scenarios and monitoring over FE models of fictitious or real structures, to produce large artificial datasets and to control all the relevant parameters that are needed to identify the weakness and the flaws of the methodology. In particular, the first case study discussed in this chapter, i.e. a concrete frame building, featured a very low complexity, thereby being used as a preliminary testbed for the approach. The tests conducted on this type of structure allowed to identify the algorithm parameters with larger influence on the performance, their range of variation and the needs of the algorithm in the calibration stage. The second case study had a slightly higher complexity in terms of geometry and structural behaviour; moreover, being calibrated on the basis of real experimental data from an existing structure, it allowed to test the methodology against more realistic scenarios.

The performed analyses did also allow to draw some conclusions regarding the damage sensitivity of different features produced by a simulated dynamic monitoring, although this aspect falls outside the

scope of the present work. In particular, in the first case study, several pairwise combinations of natural frequencies, either considering or not the ambient temperature, have been tested, showing that classifiers trained in different 2-D feature spaces have very distinct performance. Thus, it was confirmed that the identification of the quantities of interest is a fundamental step of the monitoring design, but it also suggested that such distinct sensitivity can be exploited to improve the chances of correct detection against many possible unknown damage scenarios and to extend, when possible, the classification to the higher order to include damage localisation and quantification.

The higher order classification was achieved also by following a completely different approach in the second case study. Here, two signatures were used as damage sensitive features (the PSD and the absolute value of the FFT). In this way, each measurement was represented by a set of values, namely the signature amplitude per each frequency line. The anomaly score distribution of each damage scenario was used as reference for higher order classification.

In conclusion, the numerical experiments that were discussed in the present chapter demonstrated that the damage detection strategy based on DGNSA can be effective and flexible, thus, easy to adapt to different types of structures and different level of information (number and type of features). The algorithm could correctly classify relatively small damages even based on acquisitions from sensors placed far from the damaged area. However, several factors affected the performance. Among them, the value of the algorithm parameters required a careful setting before the monitoring stage. It is worth noting that noise reduced the distance between the feature values in different conditions of the system, jeopardising the detection. This latter issue is not specific of the methodology here developed and requires a correct selection of the instruments and of the feature extraction procedures.

CHAPTER 5

Application to Field-Testing Data

Abstract

This chapter presents the application of the developed methodology to two real case studies: a small-scale masonry arch, built and tested in the laboratory of the University of Minho, and a full-scale masonry church, located within the Monastery of Jerónimos, in Lisbon. After presenting each structure, the monitoring system and the acquisition process are described. Then, the damage detection strategy is applied by testing some of the features presented in chapter 3. The experimental structure allows to assess the methodology against field recorded data from a structure subject to induced incremental damage, under controlled conditions, whereas the church of Jerónimos, even if only undamaged data are processed, allows to investigate some of the algorithm components using a large real-world case study.

5.1. Introduction

Once the methodology is validated through numerically simulated case studies, its effectiveness must be tested against field-testing data. In the present chapter, two of such applications are discussed in detail. Both of them are masonry structures. In literature, most of the case studies analysed through damage detection automatic strategies are recently designed (e.g. tall buildings, bridges, dams, offshore facilities, plants etc.) [282]–[288]. Still, the automatic damage detection could be a fundamental tool for heritage preservation. Such structural systems often present complex behaviour even in normal operational conditions and the level of knowledge regarding the materials and the components is commonly too low to address the damage identification in a model-based way. Data-driven techniques are potentially more suitable; however, it is essential to validate them in order to tackle possible shortcomings.

The two case studies discussed next allow to test the methodology for different levels of complexity, dimension and control with respect to the environmental variations. On the one side, a large-scale Manueline church located in Lisbon; on the other side, a reduced-scale brick masonry arch built in laboratorial environment. Due to the possibility to analyse data from subsequent identification campaigns over different damage scenarios induced in a controlled manner, this latter case study is exploited for a variety of purposes, thus it will be the last one to be presented.

5.2. Church of Monastery of Jerónimos, in Lisbon

5.2.1. Description of the case study and field work

The Monastery of Jerónimos, in Lisbon, is an iconic monument of the Portuguese architectural heritage. It develops around two courts, the smaller is bordered by the Sacristy, the Chapter Room, the Refectory and the Church, whereas the larger is bordered by the Ethnographic Museum of Archaeology and the Maritime Museum (Figure 5.1a). The Church, known as church of ‘Santa Maria de Belém’, is one of the most relevant examples of Late Gothic Manueline style of Portuguese architecture (Figure 5.1b). The construction started in the XVI century, following three different phases. The plan has a length of 70 m, a width of 23 m (40 m in the transept) and an average height of 24 m. It features a cruciform shape with a three-nave longitudinal limb crossed by a transept with two overlooking lateral chapels and a chancel. A single bell tower of 50 m height is placed at the corner between South and West façades. The masonry walls are made of local limestone [71]. The church presents a complex system of ribbed vaults springing directly from the external walls. Two rows of slender octagonal columns with a free height of 16 m divide the three naves (Figure 5.1). Large fan capitals spread from the head of these columns fusing with the

vaults. The thickness of the walls varies between 1.90 m in the south wall, 2.5-2.65 m in the east walls and 3.5 m in the north wall. The south wall is pierced by large openings and its stability is further ensured by three trapezoidal buttresses [71].

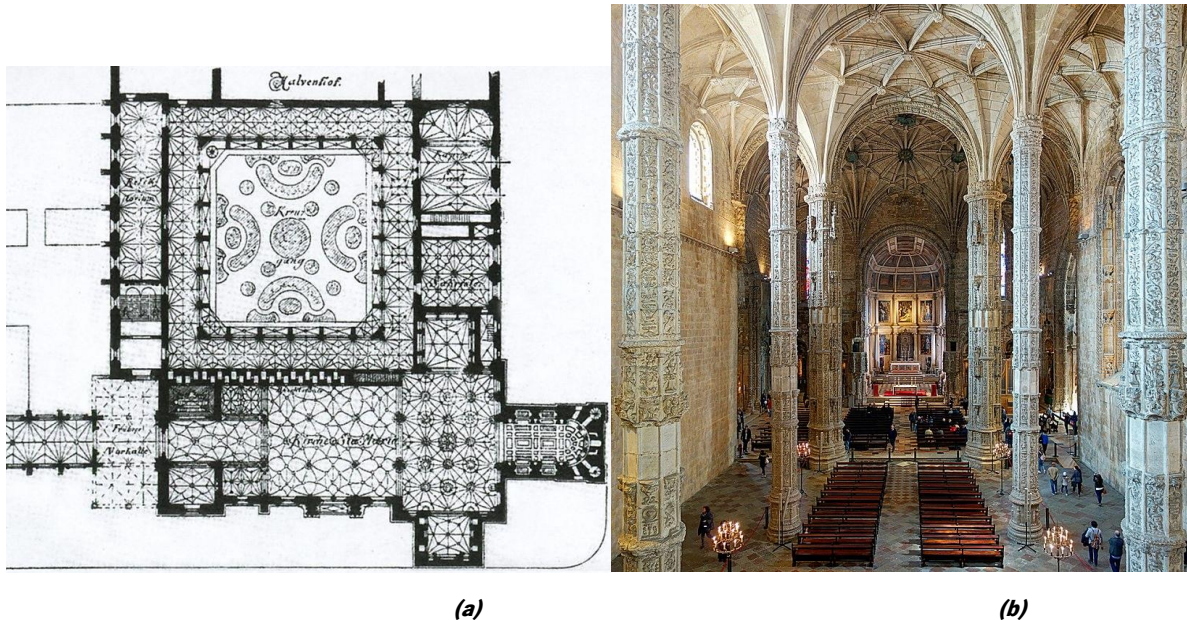


Figure 5.1: (a) Plant of the Monastery of Jerónimos by Karl Albrecht Haupt, c.1880; (b) Nave of the church. © Wikipedia Creative Commons.

Although the monastery did not register severe damages during the great earthquake of 1755, the subsequent shake of 1756 caused the collapse of one of the columns supporting the vaults and the partial collapse of the vault of the higher choir. During the XIX century, modifications were made in the structure of the two towers as well as in the roof but their effect on the structural response has not been fully investigated [71]. Due to the relevance of this monumental complex as heritage and cultural asset and its exposure to strong earthquakes, its preservation and vulnerability assessment are constant concerns. In this regard, a long term monitoring campaign was performed by the University of Minho in the main nave of the church [71], [289]. In 2005, a simplified monitoring system, composed of two strong motion recorders and two tri-axial force balance accelerometers, was installed: one accelerometer was located at the base of the structure near the chancel and connected to the master recorder, and the other accelerometer was installed on the nave extrados and connected to the slave recorder. The system was designed to perform both triggered and programmed acquisitions. As for the triggered events, a threshold level was established for the signal in order to start the acquisition whenever that level was exceeded. The programmed events consisted of monthly and seasonal acquisitions of 10 minutes, registered to follow frequency trends over time and catch possible environmental fluctuations. According to the results of a previous Operational Modal Analysis (OMA), a sampling rate of 100 Hz was adopted

for all events to digitise the signals. Temperature and humidity sensors as well as tilt meters were also deployed at critical points. The type and location of sensors were chosen according to the features to be extracted and in conformity with the directions of the Portuguese General Directorate of Cultural Heritage (DGPC), former IGESPAR, with the purpose of minimizing the visual impact of the monitoring system in the church (Figure 5.2).

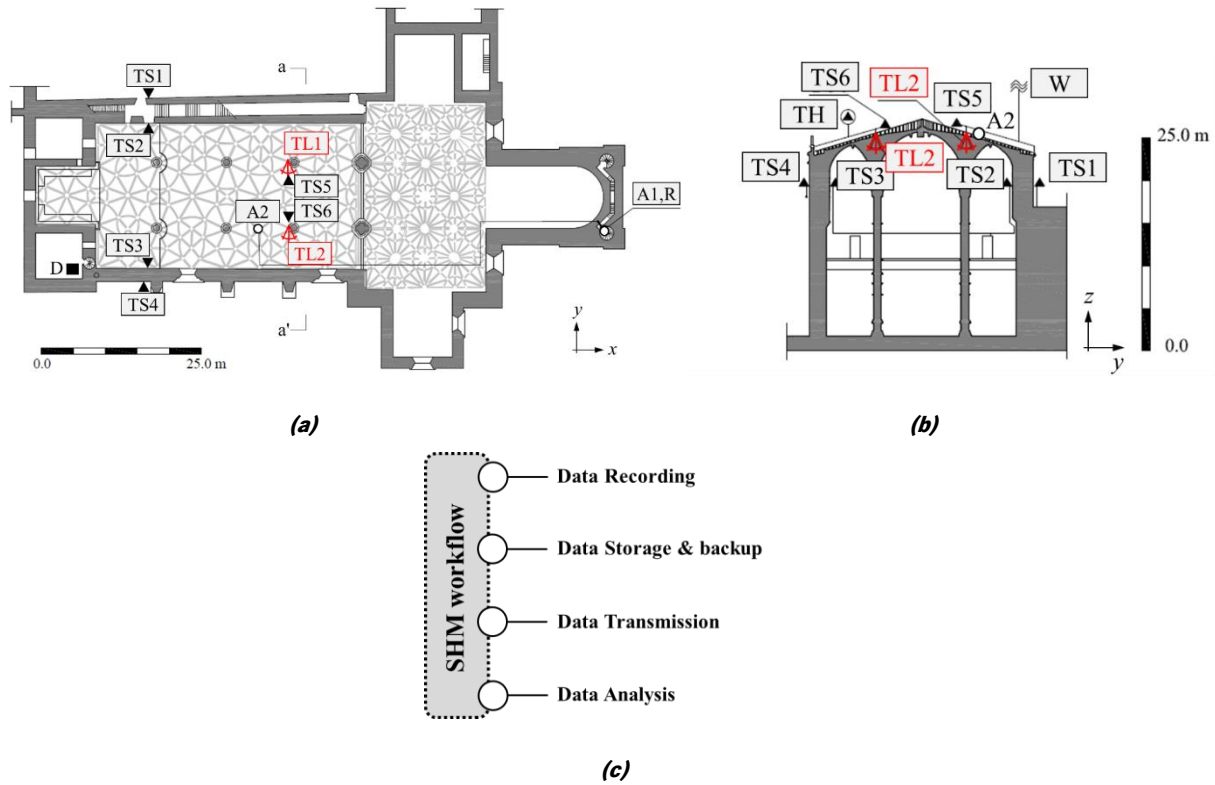


Figure 5.2: Plan (a) and cross-section (b) of the church, showing the sensor layout for the continuous monitoring: A (accelerometer), D (datalogger), R (strong motion recorder), TL (tiltmeter), TS (temperature sensor), TH (temperature and humidity sensor), W (anemometer). Schematics of the SHM data workflow (c).

The estimation of the global dynamic features of the church was performed through a MATLAB algorithm based on the Stochastic Subspace Identification method (SSI-data driven). Three main vibration modes could be clearly tracked over the monitoring period [71]. The database used in this study is composed of 788 records, measured in 255 different days between June 25th of 2005 and September 28th of 2009 (Table 5.1). The following analyses focus on the first natural frequency. As shown in Figure 5.3a, this value changes according to a cyclic behaviour which is related to the temperature variation. The correlation between ambient temperature (AT) and first natural frequency (f_1) is evident in Figure 5.3b.

Table 5.1: Synthesis of collected data.

Year	N° Days	N° samples	Period	AT	f_i
2005	11	24	25-jun/22-Nov	17.19-30.39	3.705-3.943
2006	24	111	26-Jan/27-Dec	9.09-31.83	3.639-3.939
2007	27	183	1-Jan/30-Dec	10.38-31.71	3.602-3.934
2008	17	221	1-Feb/24-Apr	12.68-25.95	3.605-3.936
2009	176	249	1-Apr/28-Sep	16.93-33.22	3.620-3.958
Tot.:	255	788	–	9.09-33.22	3.602-3.958

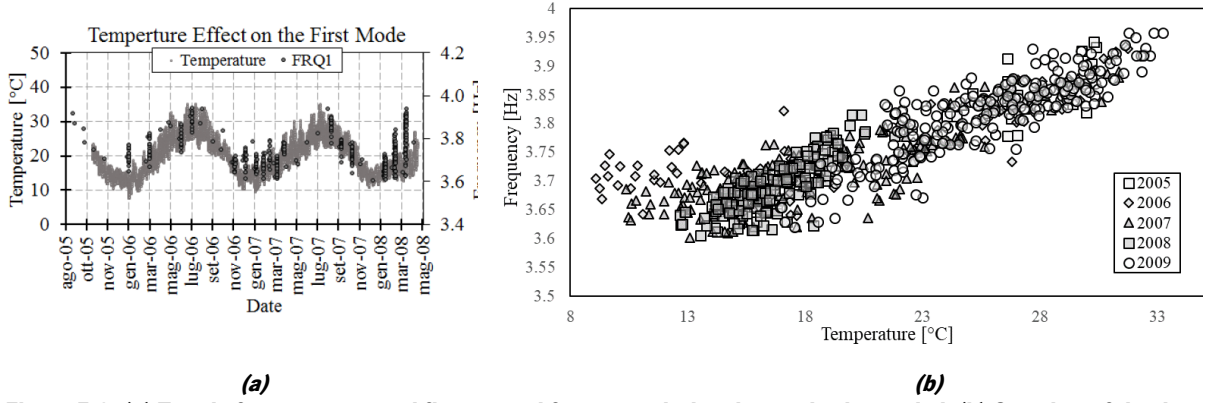


Figure 5.3: (a) Trend of temperature and first natural frequency during the monitoring period; (b) Overview of the data samples.

Although consisting of samples belonging only to the same initial condition, this dataset is used during the development of the new NSA version for a large preliminary study whose main results are presented hereafter. Part of these results has been already published in [290], [291].

5.2.2. Self set approximation

The size of the dataset allows to perform a test of the self space representation strategies presented in chapter 3, section 3.3.2. The definition of a good approximation of the system behaviour in the reference condition is a preliminary task for any classification problem, thus the results discussed next might be significant not only for the specific application of the DGNSA, but also for any anomaly detection strategy. Indeed, the result of the self set approximation is a set of points in the feature space which can be employed to approximate the possible response of the monitored system in reference condition and to train any data-driven algorithm for damage detection.

For the following analyses, it is assumed that the full self space corresponds to the area within the enveloping boundary of all the 788 samples available.

Deterministic approach

The temperature line is divided in three intervals ($<17^\circ$, 17° - 25° , $>25^\circ$) and 2, 3, 6, 12, 18, 25, 33, 40, 50 samples are drawn from each subrange, resulting in a total of 6, 9, 18, 36, 54, 75, 99, 120 and 150 samples. To assess the influence of the number of subranges, for the case of 36 and 120 samples, different divisions are analysed, respectively {3,4,6,9} and {3,4,6,8}. The definition of the intervals aims to guarantee an adequate distribution of the samples along the temperature line, especially for a small dataset.

The random generation of the samples simulates a real-world modal identification with the only condition that the range of temperature is pre-defined, furthermore ten repetitions per each number of samples are performed. No repeated samples are admitted in the same set. All these instances are compared with the exact self space in terms of intersection and similarity; moreover, for each approximation, also the nonself coverage, the approximate area and the normalised incremental area for an increasing number of samples are recorded, as described in section 3.3.2.

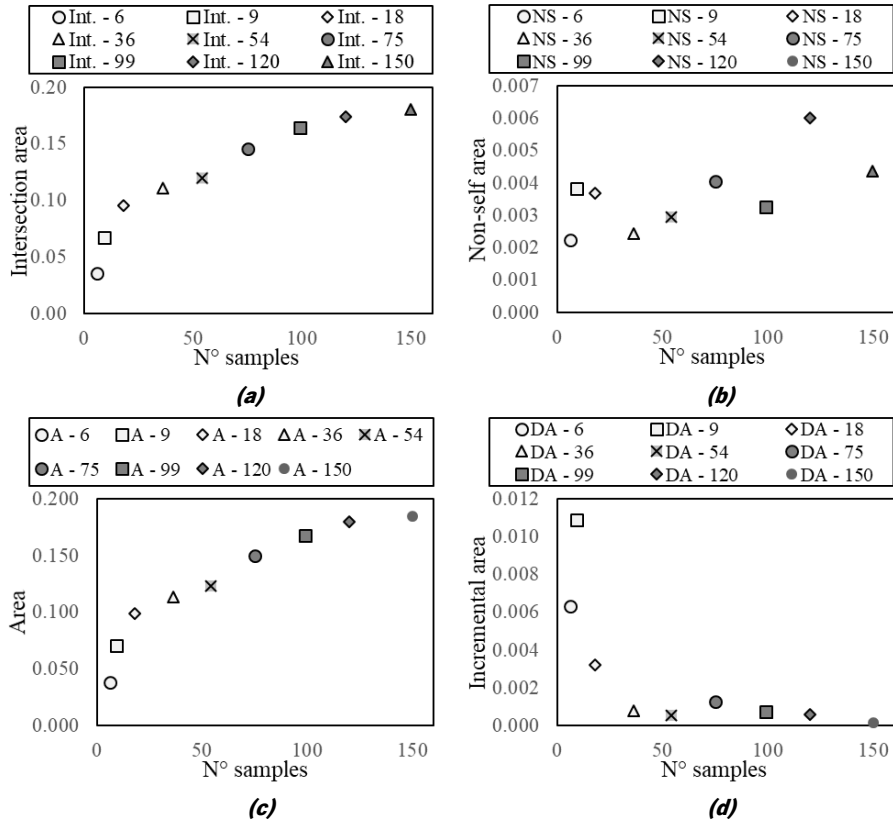
The results of the deterministic approximation are reported in Table 5.2, where “pt.” and “div.” refer to number of samples and number of divisions, respectively. Four relevant metrics, namely the intersection area, the nonself covered area, the approximate area and the incremental area value, for the different number of samples are plotted in Figure 5.4. In Figure 5.5, Figure 5.6 and Figure 5.7, some examples of the area approximated by 36, 54 and 75 samples are displayed. Figure 5.8 focuses on the comparison among different numbers of intervals. The careful analysis of these outcomes allows to draw the following preliminary conclusions:

- Similarity and intersection areas increase with the number of samples;
- Covered nonself area also increases with the number of samples, due to the higher probability of sampling boundary points, meaning that the approximated area does not fall completely inside the real one, since this is not perfectly convex. The nonself coverage has a scattered distribution affected by the number and the randomness of the sampling (Figure 5.4b). This parameter results really problem specific, thus hard to be controlled *a-priori*;
- The intersection area value has a logarithmic tendency (Figure 5.4a). The reduction of tangent slope cannot be used directly as a criterion to define an acceptable number of samples, since it requires the *a-priori* knowledge of the real self area;
- The approximate area plot and the normalised incremental area plot show an inverse trend (Figure 5.4c and d). These parameters can be used to define an acceptable number of samples, which, for the specific case, is in the range 36-75;

- Increasing the number of divisions is unlikely improving the performance for more than 4 divisions.
- This phenomenon is probably due to a forcing of the sampling in areas of the 2-D space of less interest, leaving a fewer number of samples for the dense area.

Table 5.2: Results of the deterministic approximation of the self space.

Pt.	Div.	Area Avg.	Area St.dev	Similarity mean	Similarity st.dev.	Intersection area mean	Intersection area st.dev	Nonself covered mean	Nonself covered st.dev.
ALL		0.276	–	100	–	1	–	0	–
6	3	0.071	0.032	12.7	5.0	0.036	0.064	0.0022	0.0048
9	3	0.099	0.124	23.8	8.6	0.067	0.065	0.0038	0.0062
18	3	0.113	0.090	34.2	7.5	0.096	0.053	0.0037	0.0059
36	3	0.124	0.151	39.8	11.9	0.111	0.083	0.0024	0.0027
36	4	0.123	0.110	42.4	7.3	0.119	0.111	0.0047	0.0056
36	6	0.120	0.109	42.6	6.2	0.119	0.088	0.0040	0.0034
36	9	0.123	0.122	42.3	6.3	0.118	0.132	0.0022	0.0025
54	3	0.150	0.158	43.0	6.8	0.120	0.126	0.0029	0.0035
75	3	0.168	0.205	52.0	11.0	0.146	0.194	0.0040	0.0070
99	3	0.180	0.199	59.0	8.4	0.165	0.161	0.0033	0.0032
120	3	0.196	0.212	61.7	7.0	0.174	0.155	0.0060	0.0070
120	4	0.196	0.205	67.0	5.9	0.189	0.166	0.0068	0.0053
120	6	0.180	0.200	67.3	4.7	0.190	0.193	0.0059	0.0027
120	8	0.184	0.150	62.7	5.4	0.176	0.182	0.0042	0.0036
150	3	0.038	0.181	64.2	5.6	0.180	0.157	0.0043	0.0023

**Figure 5.4: Trend of the Intersection area (a), the nonself covered area (b), the approximate area (c) and the normalised incremental area (d) for different number of samples.**

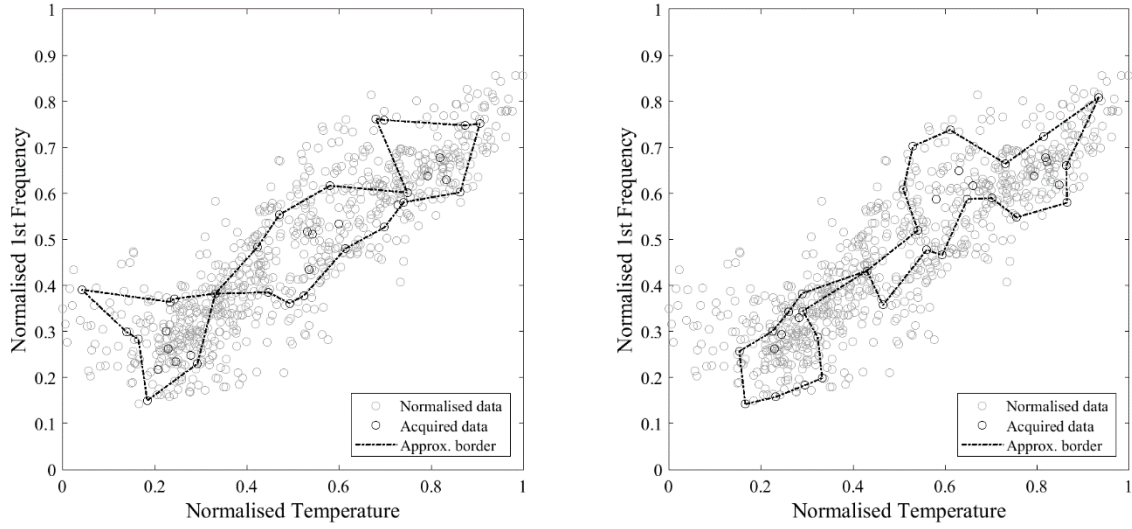


Figure 5.5: Approximate areas by mean of 36 samples compared with the whole dataset in two out of the ten tested repetitions.

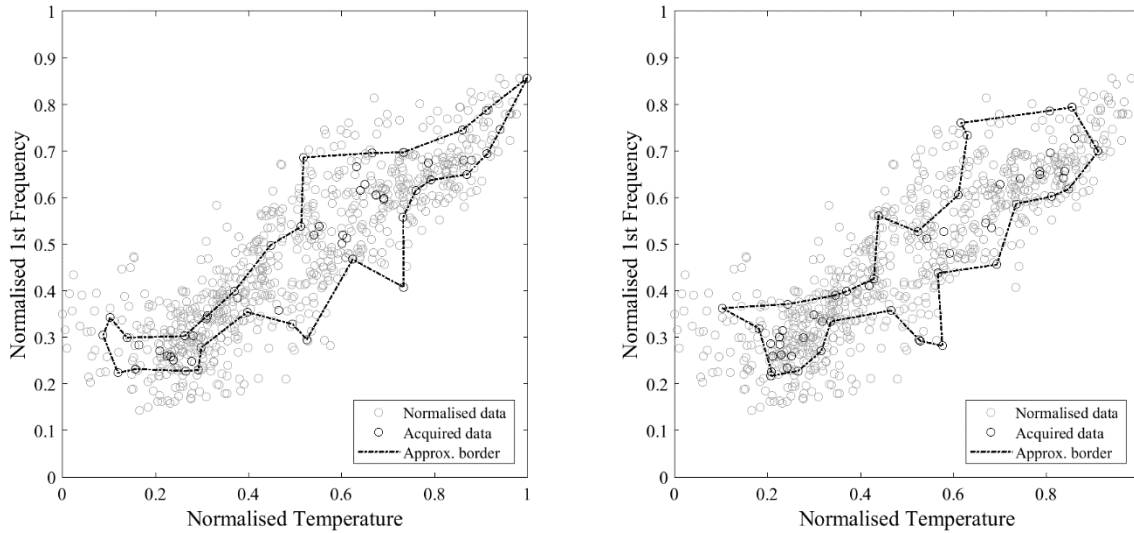


Figure 5.6: Approximate areas by mean of 54 samples compared with the whole dataset in two out of the ten tested repetitions.

To test the reliability of the incremental growth, a single monitoring instance is simulated, starting from an initial set of 6 samples and adding one record per time up to 75. The resulting values of area and normalised increment are reported in Figure 5.9. The general trend confirms the previous results. Sudden drops in the area value emerge, due to an excessive shrinkage of the boundary caused by the automatic procedure. In Figure 5.10, an example of such phenomenon moving from 54 to 67 samples is pointed out through arrows. Such issue can be easily solved postprocessing the automatic identified area.

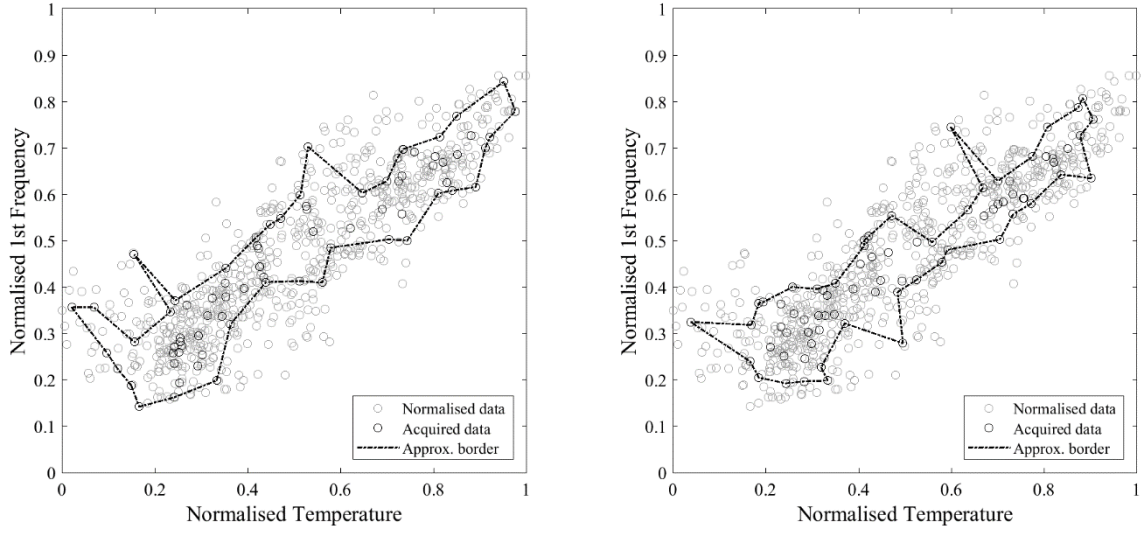


Figure 5.7: Approximate areas by mean of 75 samples compared with the whole dataset in two out of the ten tested repetitions.

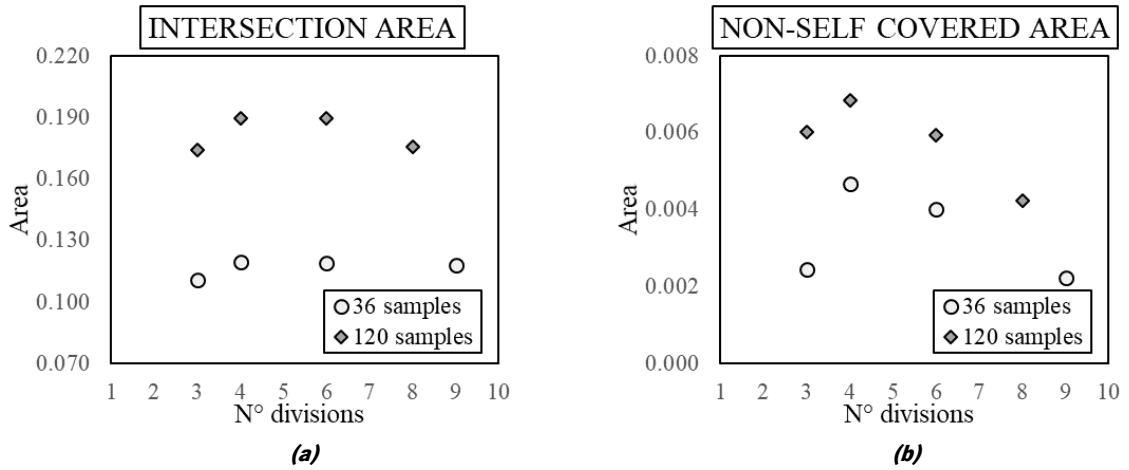


Figure 5.8: Trend of the intersection area (a) and the non-self covered area (b) for different number of divisions and constant data samples.

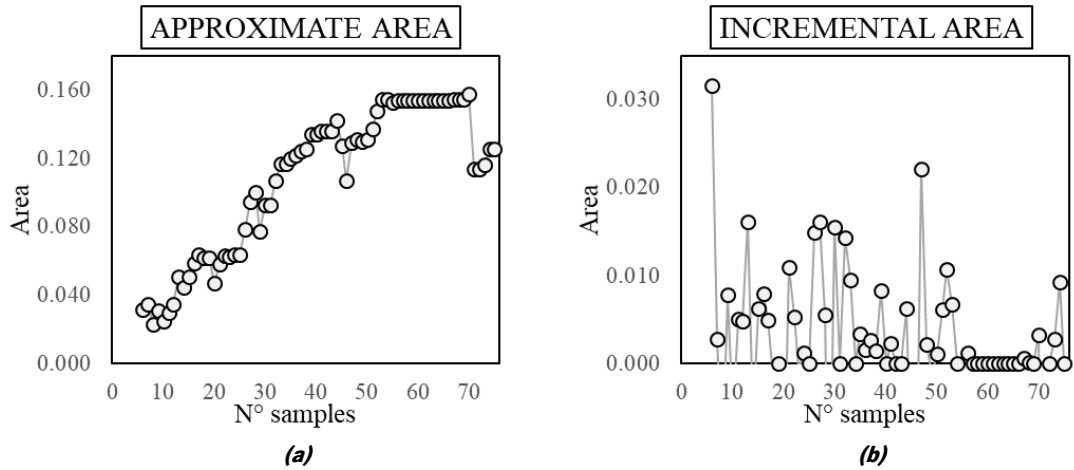


Figure 5.9: Trend of the approximate area (a) and the normalised incremental area (b) for different number of samples adding at each step a single new record to the previous set.

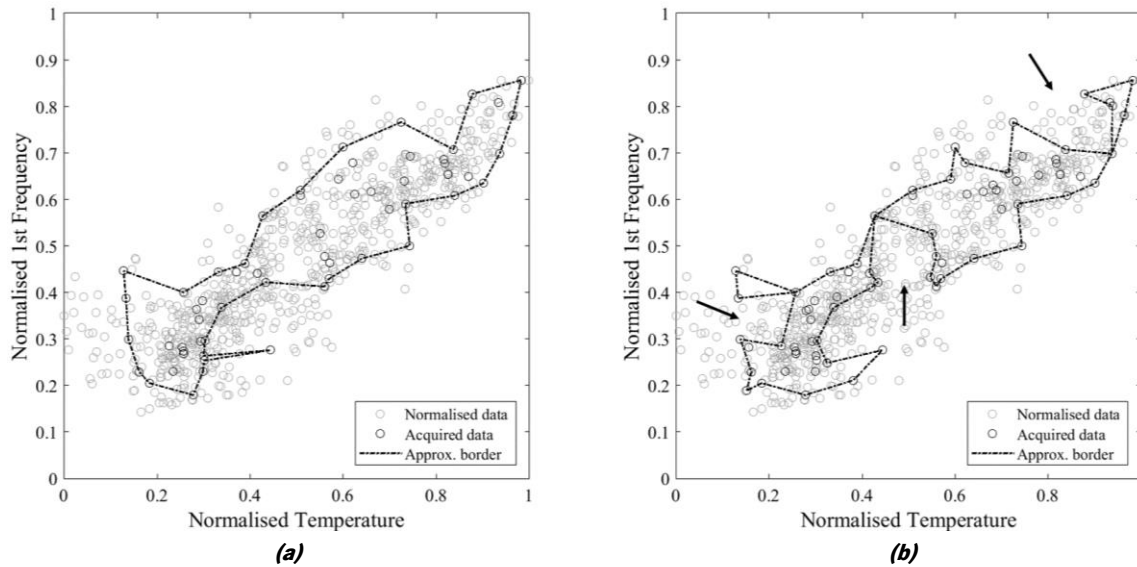


Figure 5.10: Approximate area provided by 54 (a) and 67 (b) samples.

Finally, based on the available data, a possible plan for a data campaign is presented. Although the aim is to identify 60 records, 20 per each subrange of temperature ($<17^\circ$, 17° - 25° , $>25^\circ$), based on the available data, the closest instance consists of 63 samples almost evenly distributed (18-27-18) in the three subranges (Figure 5.11, Table 5.3). The 63 samples are collected in 6 days with hourly acquisitions from 9 a.m. to 17 p.m. between February the 1st and July the 1st. Since the monitoring is only simulated and a continuous monitoring is not available, the days do not belong to the same year. However, they are indicative of the actual temperature in the corresponding period of the year.

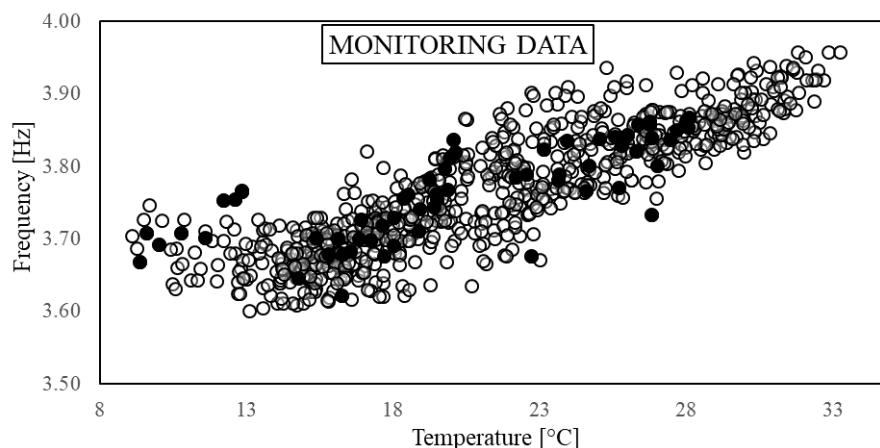


Figure 5.11: Self dataset and monitored data campaign.

The trends of the approximate area and of the normalised incremental area are reported in Figure 5.12. As seen in the case of Figure 5.9, the automatic procedure produces sudden variations in the trends (Figure 5.12 a and d). Thus, a manual modification of the approximate area is performed, achieving a smoother trend (Figure 5.12 b and e). The trend allows to identify a first small jump due to acquisitions

collected from the second range of the temperature, during the second day. A second significant jump appears when acquisitions from the third range are added. After each jump the incremental area reduces significantly as well as the tangent slope of the approximate area. Finally, the average daily trend is introduced (Figure 5.12 c and f) as a condensed index of the approximation quality.

Table 5.3: Designed data campaign with results. Three colours are used to distinguish the three ranges.

Day	Hour	Temp.	Freq.1	Freq.3	Freq.4	Day	Hour	Temp.	Freq.1	Freq.3	Freq.4
01/02	09:00	9.35	3.669	6.292	7.270	01/06	09:00	22.71	3.677	6.510	7.409
01/02	10:00	9.59	3.708	6.272	7.256	01/06	10:00	23.11	3.823	6.513	7.400
01/02	11:00	10.01	3.693	6.299	7.281	01/06	11:00	23.9	3.836	6.535	7.400
01/02	12:00	10.75	3.708	6.282	7.283	01/06	12:00	25.03	3.838	6.542	7.409
01/02	13:00	11.58	3.701	6.295	7.304	01/06	13:00	25.55	3.841	6.567	7.460
01/02	14:00	12.2	3.753	6.320	7.275	01/06	14:00	25.98	3.844	6.586	7.460
01/02	15:00	12.61	3.755	6.340	7.286	01/06	15:00	26.35	3.858	6.581	7.468
01/02	16:00	12.81	3.767	6.358	7.309	01/06	16:00	26.75	3.858	6.623	7.479
01/02	17:00	12.84	3.765	6.358	7.336	01/06	17:00	26.74	3.862	6.588	7.477
02/03	09:00	14.76	3.646	6.271	7.160	03/06	09:00	22.18	3.786	6.417	7.335
02/03	10:00	15.36	3.701	6.286	7.279	03/06	10:00	22.54	3.790	6.447	7.334
02/03	11:00	16.1	3.701	6.308	7.269	03/06	11:00	23.61	3.781	6.453	7.352
02/03	12:00	16.89	3.727	6.359	7.274	03/06	12:00	24.67	3.801	6.532	7.368
02/03	13:00	17.69	3.677	6.418	7.302	03/06	13:00	25.77	3.829	6.588	7.414
02/03	14:00	18.5	3.761	6.464	7.314	03/06	14:00	26.81	3.840	6.573	7.437
02/03	15:00	19.21	3.782	–	–	03/06	15:00	27.66	3.850	6.620	7.464
02/03	16:00	19.75	3.797	–	–	03/06	16:00	28.05	3.855	6.629	7.474
02/03	17:00	19.92	3.811	–	–	03/06	17:00	28.07	3.867	6.655	7.474
03/03	09:00	15.77	3.680	6.254	7.204	01/07	09:00	23.64	3.790	6.398	7.313
03/03	10:00	16.27	3.680	6.286	7.242	01/07	10:00	24.56	3.766	6.404	7.312
03/03	11:00	16.82	3.701	6.324	7.350	01/07	11:00	25.67	3.771	6.433	7.310
03/03	12:00	17.62	3.720	6.368	7.316	01/07	12:00	26.26	3.821	6.455	7.350
03/03	13:00	18.02	3.729	6.395	7.343	01/07	13:00	26.8	3.734	6.500	7.351
03/03	14:00	18.35	3.757	6.405	7.301	01/07	14:00	27	3.802	6.514	7.395
03/03	15:00	18.91	3.742	–	–	01/07	15:00	27.43	3.837	6.539	7.403
03/03	16:00	19.46	3.763	–	–	01/07	16:00	27.92	3.855	6.543	7.403
03/03	17:00	19.47	3.755	–	–	01/07	17:00	27.97	3.864	6.566	7.442
04/03	09:00	16.23	3.622	6.265	7.265						
04/03	10:00	16.52	3.683	6.284	7.261						
04/03	11:00	17.22	3.699	6.317	7.263						
04/03	12:00	18.02	3.691	6.377	7.275						
04/03	13:00	18.86	3.710	6.428	7.293						
04/03	14:00	19.39	3.744	6.459	7.309						
04/03	15:00	19.85	3.768	–	–						
04/03	16:00	20.1	3.819	–	–						
04/03	17:00	20.03	3.837	–	–						

Colour key:

 Range $T < 17^{\circ}$	 Range $17^{\circ} < T < 25^{\circ}$	 Range $T > 25^{\circ}$
--	---	--

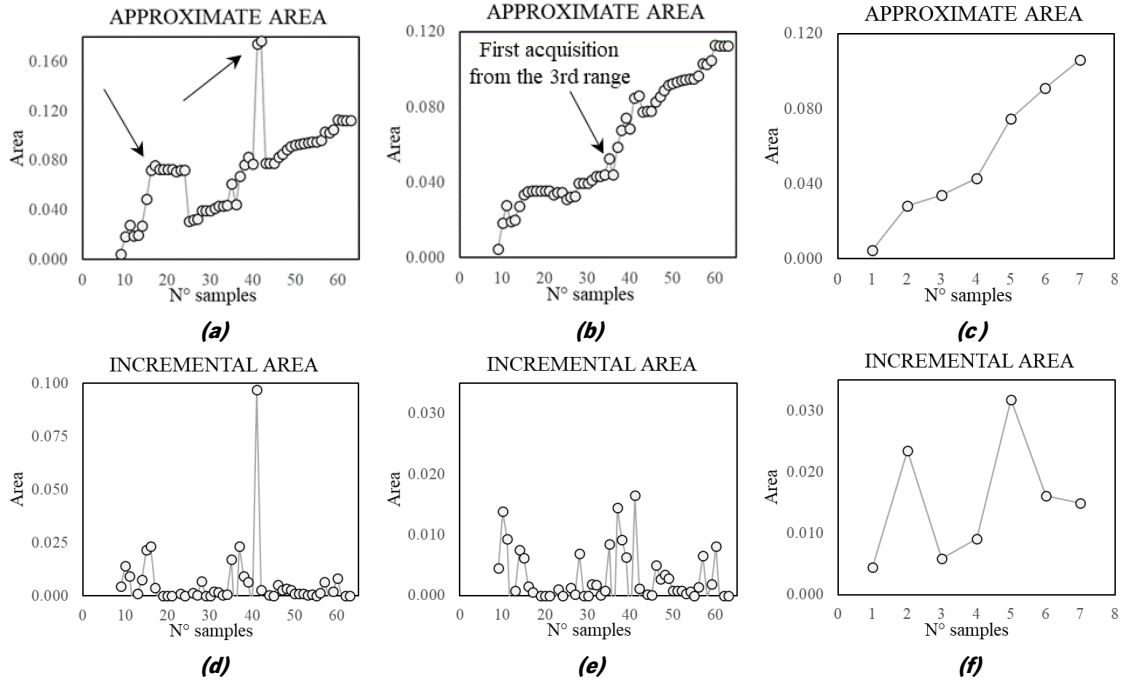


Figure 5.12: Trends of the approximate area and incremental area considering the automatic procedure (a-d), manual correction of some approximation (b-e) and daily average (c-f).

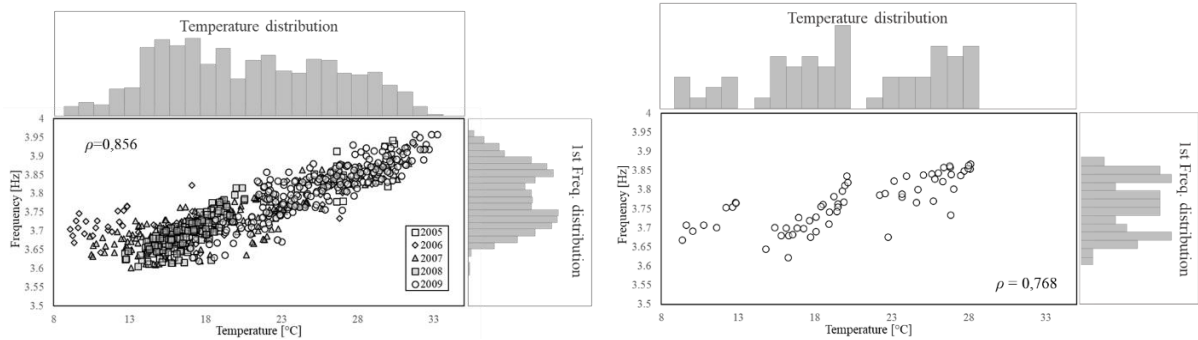


Figure 5.13: Overview of the data samples. Comparison between the whole dataset and the simulated monitoring campaign with reduced samples.

The comparison between the whole dataset and the simulated monitoring data is shown qualitatively in Figure 5.13, where the whole sample is subdivided by years. As for the monitoring data, the Pearson coefficient is also highlighted. Table 5.4 reports the statistics for both sets.

It is worth noting that, even though the distribution of the monitored data is not completely representative of the population's distribution, the Pearson coefficient values are very close, suggesting that the feature selection could be performed on the reduced dataset with similar results. The differences in the distribution are likely affected by a lack of samples in the highest range of temperature. The maximum value of the population is indeed 5° C greater than in the monitored data. This lack might affect the interquartile distance, that is smaller for the monitored data, and the skewness that becomes negative since the right tail is missing. Nonetheless, the measures of central tendency are properly approximated. Also, for the natural frequencies the extreme values are missing, even though upper and lower ranges

which are cut off by the sampling are comparable. Likewise for the temperature, the main difference is in the skewness.

Table 5.4: Overview of the statistics. Comparison between the whole and reduced (simulated monitoring) dataset.

	Whole dataset		Reduced dataset			
	T	f_i	T	ΔT	f_i	Δf_i
Max:	33.22	3.96	28.07	-15.5%	3.87	-2.3%
Min:	9.09	3.53	9.35	2.9%	3.62	2.7%
Mean:	21.16	3.77	20.31	-4.0%	3.77	$\sim 0\%$
St. Dev.:	5.64	0.09	5.42	-3.9%	0.07	-24.4%
Skewness:	0.15	0.10	-0.25	-271.2%	-0.14	-245.4%
Kurtosis:	-0.99	-1.01	-0.94	-5.0%	-1.05	4.2%
Median:	20.73	3.76	19.75	-4.7%	3.77	0.1%
Interquartile:	9.45	0.14	8.94	-5.4%	0.12	-16.9%
Max. daily range:	7.14	0.20	–	–	–	–
Min. daily range:	1.27	0.04	–	–	–	–

Stochastic approach

The three stochastic algorithms for the approximation, presented in section 3.3.2, namely affinity method, Gaussian method and B-NN-ASR algorithm, are compared hereafter. Particularly, two different values for R (e.g. the radius value for the max. affinity density sample in the affinity method or for the max. probability density sample in the Gaussian method) and for k (number of nearest neighbours in the B-NN-ASR algorithm) are considered, namely $R = 0.1$, $R = 0.2$, $k = 4$ and $k = 6$. Furthermore, due to the characteristics of the sample set, the value of α , which [253] recommends equal to 150° , is reduced to 110° . The algorithms are validated by using a set of 326 artificial nonself elements presented in Figure 5.14.

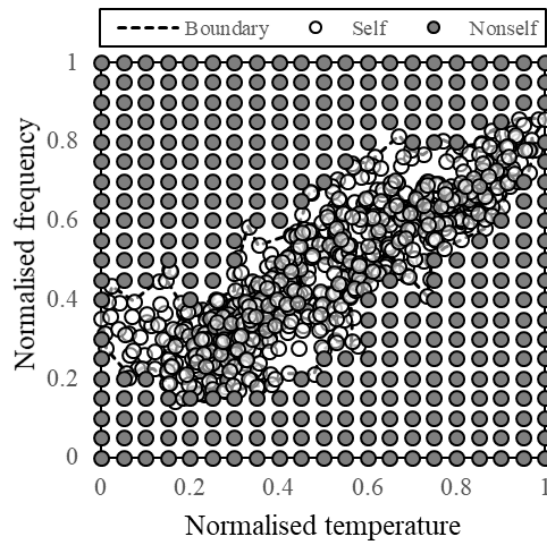


Figure 5.14: Representation of the full self set with its boundary and the artificial non-self set used for performance assessment.

Self and nonself coverage are formulated as follows:

$$SC = \frac{N^{\circ} \text{ of self samples covered by the approximate self set}}{\text{Total } N^{\circ} \text{ of self samples}} \quad (Eq. 5.41)$$

$$NSC = \frac{N^{\circ} \text{ of nonself samples covered by the approximate self set}}{\text{Total } N^{\circ} \text{ of nonself samples}}$$

Their values are reported in Table 5.5 and Table 5.6, respectively. The results are also represented in Figure 5.15. The following aspects can be highlighted:

- Affinity and Gaussian approaches produce increasing self and nonself coverages with increasing number of samples;
- Given the same R , the Gaussian method outperforms the affinity method, presenting a lower nonself coverage for the same self coverage, thus being less prone to false negative errors. However, to achieve the same self coverage, the Gaussian approach requires more samples, especially for a smaller R ;
- When a larger radius R is adopted, less samples are required to achieve a good coverage, however, the nonself area covered becomes quickly significant. A smaller radius, even if more conservative, leads to the presence of holes within the approximate self area, which might induce false positive errors. This does not happen in deterministic approximation since all the area within the samples is considered as belonging to self;
- Among the three methodologies, the B-NN-ASR algorithm showed a completely different behaviour, proving to be less effective. Due to the characteristics of the training set, the algorithm fails to identify the inner samples and, consequently, the boundary samples are not moved toward an internal neighbour. For the same reason, the size of the holes within the approximate self space is much lower, but the nonself area covered is too large.

Such results can be qualitatively seen in Figure 5.16.

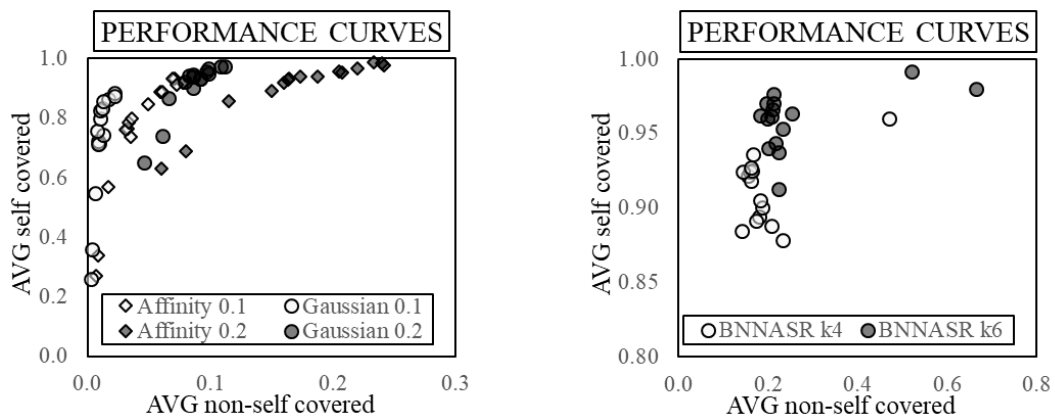


Figure 5.15: Performance curves of the different stochastic methods considering different sample sets.

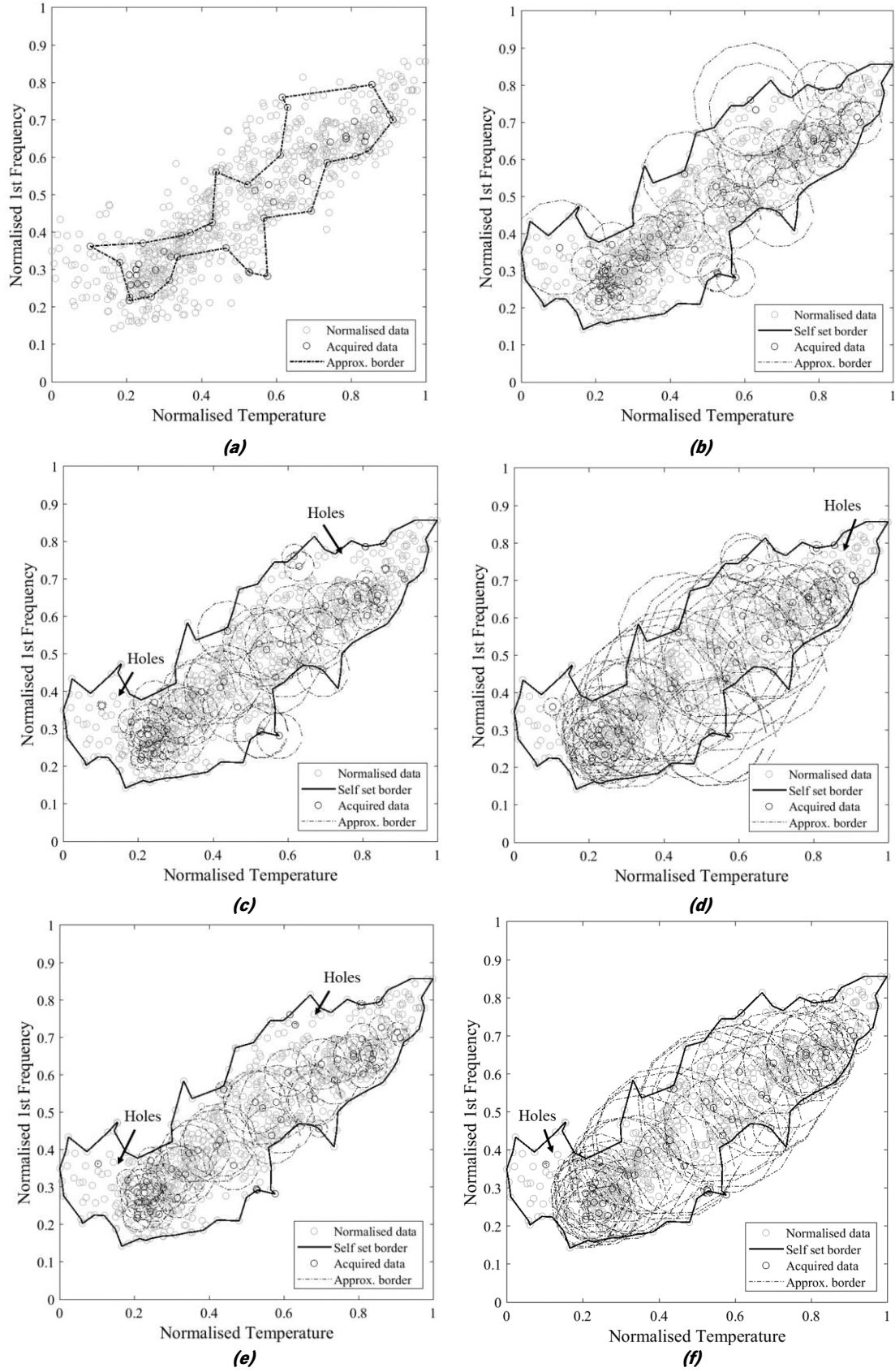


Figure 5.16: 54 points approximate areas: (a) deterministic approximation; (b) B-NN-ASR k4; (c) affinity 0.1; (d) affinity 0.2; (e) Gaussian 0.1; (f) Gaussian 0.2.

Table 5.5: Average self set coverage of the different stochastic approximations.

Pt.	Div.	Affinity 0.1	Affinity 0.2	Gaussian 0.1	Gaussian 0.2	B-NN-ASR k4	B-NN-ASR k6
6	3	0.269	0.629	0.259	0.649	0.960	0.980
9	3	0.337	0.688	0.359	0.740	0.878	0.992
18	3	0.567	0.857	0.547	0.866	0.888	0.913
36	3	0.737	0.891	0.716	0.903	0.894	0.938
36	4	0.765	0.930	0.722	0.934	0.900	0.953
36	6	0.762	0.934	0.713	0.932	0.892	0.944
36	9	0.785	0.941	0.742	0.949	0.905	0.963
54	3	0.800	0.919	0.758	0.922	0.884	0.940
75	3	0.846	0.940	0.798	0.940	0.922	0.961
99	3	0.887	0.954	0.826	0.944	0.924	0.960
120	3	0.889	0.957	0.832	0.947	0.918	0.966
120	4	0.918	0.977	0.864	0.965	0.936	0.977
120	6	0.933	0.984	0.883	0.975	0.925	0.970
120	8	0.931	0.987	0.874	0.972	0.925	0.962
150	3	0.913	0.965	0.856	0.957	0.927	0.970

Table 5.6: Average nonself set coverage of the different stochastic approximations.

Pt.	Div.	Affinity 0.1	Affinity 0.2	Gaussian 0.1	Gaussian 0.2	B-NN-ASR k4	B-NN-ASR k6
6	3	0.007	0.060	0.002	0.046	0.471	0.664
9	3	0.009	0.080	0.004	0.061	0.231	0.521
18	3	0.017	0.115	0.006	0.066	0.206	0.224
36	3	0.035	0.150	0.009	0.086	0.179	0.224
36	4	0.033	0.163	0.008	0.092	0.186	0.233
36	6	0.031	0.164	0.009	0.092	0.173	0.215
36	9	0.033	0.174	0.013	0.099	0.182	0.252
54	3	0.036	0.160	0.008	0.079	0.141	0.201
75	3	0.049	0.187	0.010	0.086	0.154	0.208
99	3	0.059	0.207	0.010	0.083	0.158	0.198
120	3	0.061	0.205	0.012	0.086	0.161	0.209
120	4	0.077	0.241	0.017	0.099	0.166	0.212
120	6	0.071	0.240	0.021	0.109	0.164	0.213
120	8	0.069	0.233	0.022	0.112	0.142	0.182
150	3	0.072	0.220	0.013	0.097	0.163	0.197

5.2.3. Influence of the DGNSA qualitative and quantitative parameters

The available real-world dataset allows to test also the influence of the algorithm parameters on the overall performance of the new DGNSA. The feature space considered in this case is given by the temperature and the first frequency, both normalised through the min-max normalisation algorithm. The following parameters are tested in different subsequent numerical experiments:

- Self radius r_{self} ;
- Detector radius r_{det} ;
- “Moving away” process through the maturity age t ;
- Censoring distance formulation.

An initial pilot study is performed to define their range of variability. Moreover, the algorithm performance is further assessed considering the factors reported below:

- Performance metrics;
- Number of samples in the training set;
- DOE strategy.

At this stage, since the dataset does not include samples from the system in changed conditions, a strategy for the definition of artificial nonself samples is developed and tested in order to use them for the algorithm validation.

The first set of experiment consists in a full factorial design considering three levels of detector radius and maturity age, and three different formulations of the censoring distance. Self radius r_{self} was fixed to 0.05 and adaptation rate η_0 to 0.005. Three hundred samples are randomly selected for the training stage. To analyse the effect of randomness, six repetitions for each combination of levels are carried out. The feasibility of a performance metric based only on the number of detectors (performance in terms of computational effort) and on the number of false positives (performance in terms of quality of the solution) is explored. For this purpose, a weighted objective function which gives more importance to the FP (75%) and less to the number of detectors (25%) is used as it is better to compromise on the memory allocation than on the error reduction:

$$f = 0.25 \frac{\text{detectors}}{1423} + 0.75 \frac{FP}{269} \quad (\text{Eq. 5.42})$$

It is noted that the percentage weights is arbitrarily chosen. The values are normalized to the highest values found during the experiment (e.g. 1423 detectors and 269 FP).

The best combination is given by a censoring distance formulated as $(r_{self} + r_{det})$, a detector radius of 0.0707 (10 divisions) and a maturity age of 45. The same best combination is achieved by testing different numbers of samples and self radii.

To assess the quality of the performance metric above, the optimised classifier is tested over artificially generated data, namely:

- 1000 healthy data, randomly generated according to the joint probability mass function of the two variables (temperature and frequency) discretized into intervals of 2°C and 0.02 Hz, respectively;
- 520 outliers, 40 per each interval of 2°C, randomly generated according to the marginal distribution of the frequencies into the temperature intervals.

The new simulated data are normalized to the same range of the data acquired and all the outliers normalized to values bigger than 1 or lower than 0 are rejected. The following conclusions are drawn:

- A performance metric based only on self data rewards classifiers with low false positive rate (1%) but, consequently, high false negative rate (21%);
- The area under the ROC curve, considering actual self and actual nonself samples, is a more informative metric.

Therefore, in the following experiment, the area under the ROC curve is used to assess the performance. In this second campaign, an adaptation rate η_0 equal to 0.005 is chosen, whereas all the affecting parameters are tested at three levels, reported in Table 5.7. Three levels are also defined for the number of training samples, namely 50, 150 and 300 and six repetitions are carried out to analyse the effect of randomness. These corresponds to 18 tests, each composed of 81 combinations of factor's levels. The exact value of the parameters in the 81 combinations is reported in Appendix (Table A1).

Table 5.7: Parameter levels.

Levels	n	t	$dist$	r_{det}	r_{self}
1	70	15	r_{det}	0.0101	0.01
2	35	30	$0.5(r_{self} + r_{det})$	0.0202	0.03
3	10	45	$(r_{self} + r_{det})$	0.0707	0.07

Each repetition is based on a different training set, randomly sampled from the full dataset. For the validation, the same self and nonself artificial sets are used against each instance (Figure 5.17):

- 470 health-state data;
- 470 outliers.

These datasets are generated by considering the joint probability mass function of the two variables (temperature and frequency), respectively discretized into intervals of 2°C and 0.02 Hz, sampling the same number of data for each interval. However, all the samples that exceed the bounds are rejected.

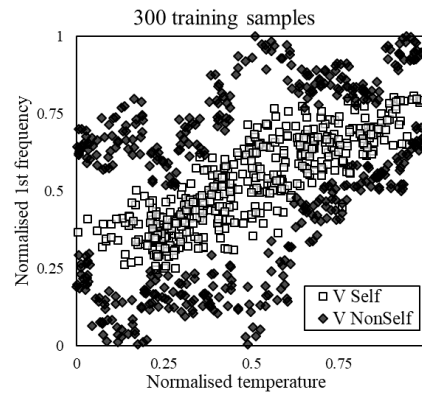


Figure 5.17: Validating self data and validating nonself data in the unitary feature space.

Table 5.8: Best combination and performance metrics per each test performed. The name of the test refers to the number of training samples and the repetition. In some tests, more combinations had the same best score. Regarding the number of FP and FN it is worth remembering that the total real negatives and real positives are 470. The three best overall performances (for 50, 150 and 300 training samples) are highlighted in bold and grey.

Test	dist	r_det	r_self	t	FP	FN	ROCAUC	Test	dist	r_det	r_self	t	FP	FN	ROCAUC
50-R1	H	L	H	H	25	51	0.919	300-R1	I	L	H	L	25	45	0.926
50-R2	H	L	H	L	17	36	0.944	300-R1	I	L	H	I	25	45	0.926
50-R3	H	L	H	L	23	53	0.919	300-R1	H	L	I	L	25	45	0.926
50-R4	H	I	H	L	14	60	0.921	300-R1	H	L	I	I	25	45	0.926
50-R4	H	I	H	I	14	60	0.921	300-R2	H	L	I	L	17	37	0.943
50-R5	H	L	H	L	12	66	0.917	300-R2	H	L	I	I	17	37	0.943
50-R5	H	L	H	I	12	66	0.917	300-R2	H	L	I	H	17	37	0.943
50-R5	H	L	H	H	12	66	0.917	300-R3	I	L	H	L	15	44	0.937
50-R6	H	L	H	L	25	59	0.911	300-R3	I	L	H	I	15	44	0.937
50-R6	H	L	H	I	25	59	0.911	300-R3	I	L	H	H	15	44	0.937
50-R6	H	L	H	H	25	59	0.911	300-R3	H	L	I	I	15	44	0.937
Test	dist	r_det	r_self	t	FP	FN	ROCAUC	300-R3	H	L	I	L	15	44	0.937
150-R1	I	I	H	H	37	18	0.941	300-R3	H	L	I	H	15	44	0.937
150-R2	I	I	H	I	45	32	0.918	300-R4	I	L	H	L	18	40	0.938
150-R2	I	I	H	H	45	32	0.918	300-R4	I	L	H	I	18	40	0.938
150-R3	I	I	H	I	29	33	0.934	300-R4	H	L	I	L	18	40	0.938
150-R3	I	I	H	H	29	33	0.934	300-R4	H	L	I	I	18	40	0.938
150-R3	H	I	I	I	22	40	0.934	300-R5	H	I	I	L	13	40	0.944
150-R4	H	I	I	L	29	27	0.940	300-R5	H	I	I	I	13	40	0.944
150-R4	H	I	I	I	30	26	0.940	300-R5	H	I	I	H	13	40	0.944
150-R4	H	I	I	H	29	27	0.940	300-R6	I	I	H	L	13	29	0.955
150-R5	H	I	I	L	40	28	0.928	300-R6	I	I	H	I	13	29	0.955
150-R6	H	I	I	L	31	44	0.920	300-R6	I	I	H	H	13	29	0.955

Table 5.8 and Table 5.9 report the best combination given by each test with the corresponding performance score. The following conclusions can be drawn:

- Irrespective of the other factors (number of samples, censoring distance, self and detector radii), the maturity age has an almost negligible influence on the performance;
- The best censoring distance is almost always formulated as $(r_{self} + r_{det})$ or, less commonly, as $0.5(r_{self} + r_{det})$. In any case, its value is more than twice the value of the detector radius.
- The best detector radius is commonly equal to the low level, and it is never equal to the high level (Table 5.8);
- The best self radius is commonly equal to the high level and it is never equal to the low level. In any case it is larger than the detector radius (Table 5.8);

- When less training samples are available, larger censoring distance and self radius are required (Table 5.8). This is expected as less samples are available to cover the whole self set;
- The worst performances are given by a detector radius larger than the censoring distance. Due to the formulation of the distance, this can happen with $0.5(r_{self} + r_{det})$, when $r_{self} < r_{det}$. In this situation, not only the small self radius does not account for self points that are not known during the training, but also the detectors are moved away from the known self points to a distance smaller than the distance that triggers them during the monitoring (Table 5.8);
- The performance, on average, increases with the number of training samples. The variability in the performance, changing the parameter setting, reduces by increasing the training samples. A lower variability leads to more combination with the same score of the performance metric (Table 5.8). Therefore, the smaller the available training set and the more important both the proper algorithm setting and the classifier comparison and validation are.

Table 5.9: Same of Table 5.8 but with explicit values.

Test	<i>dist</i>	<i>r_{det}</i>	<i>r_{self}</i>	<i>t</i>	Test	<i>dist</i>	<i>r_{det}</i>	<i>r_{self}</i>	<i>t</i>
50-R1	0.0801	0.0101	0.07	45	300-R1	0.0401	0.0101	0.07	15
50-R2	0.0801	0.0101	0.07	15	300-R1	0.0401	0.0101	0.07	30
50-R3	0.0801	0.0101	0.07	15	300-R1	0.0401	0.0101	0.03	15
50-R4	0.0902	0.0202	0.07	15	300-R1	0.0401	0.0101	0.03	30
50-R4	0.0902	0.0202	0.07	30	300-R2	0.0401	0.0101	0.03	15
50-R5	0.0801	0.0101	0.07	15	300-R2	0.0401	0.0101	0.03	30
50-R5	0.0801	0.0101	0.07	30	300-R2	0.0401	0.0101	0.03	45
50-R5	0.0801	0.0101	0.07	45	300-R3	0.0401	0.0101	0.07	15
50-R6	0.0801	0.0101	0.07	15	300-R3	0.0401	0.0101	0.07	30
50-R6	0.0801	0.0101	0.07	30	300-R3	0.0401	0.0101	0.07	45
50-R6	0.0801	0.0101	0.07	45	300-R3	0.0401	0.0101	0.03	15
150-R1	0.0451	0.0202	0.07	45	300-R3	0.0401	0.0101	0.03	30
150-R2	0.0451	0.0202	0.07	30	300-R3	0.0401	0.0101	0.03	45
150-R2	0.0451	0.0202	0.07	45	300-R4	0.0401	0.0101	0.07	15
150-R3	0.0451	0.0202	0.07	30	300-R4	0.0401	0.0101	0.07	30
150-R3	0.0451	0.0202	0.07	45	300-R4	0.0401	0.0101	0.03	15
150-R3	0.0502	0.0202	0.03	30	300-R4	0.0401	0.0101	0.03	30
150-R4	0.0502	0.0202	0.03	15	300-R5	0.0502	0.0202	0.03	15
150-R4	0.0502	0.0202	0.03	30	300-R5	0.0502	0.0202	0.03	30
150-R4	0.0502	0.0202	0.03	45	300-R5	0.0502	0.0202	0.03	45
150-R5	0.0502	0.0202	0.03	15	300-R6	0.0451	0.0202	0.07	15
150-R6	0.0502	0.0202	0.03	15	300-R6	0.0451	0.0202	0.07	30
					300-R6	0.0451	0.0202	0.07	45

Finally, a third set of experiments is carried out in order to assess the influence of the DOE strategy on the definition of the best classifier. The maturity age t is set to 30, and the adaptation rate η_0 is set to 0.005. The influence of the other parameters is instead analysed through two different designs of the experiments. One is a three-level full factorial statistical design, whereas the other one is a central composite design. The central composite design is defined only for two parameters, r_{det} and r_{self} , and it is repeated for the same three levels of the censoring distance. Compared to the factor levels used in the three-level full factorial design, for the CCD, the factors are scaled down according to the so-called inscribed central composite design (Figure 3.11). Therefore, the star points are located at the maximum and the minimum levels of the full factorial design. This is essentially done to compare the two approaches in terms of achieved response surfaces, without extending the limit of the levels to avoid too large or too small radii.

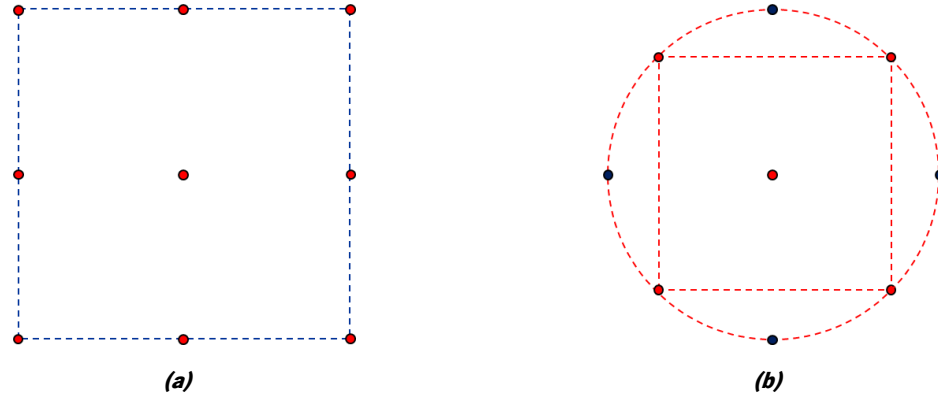


Figure 5.18: Comparison between full factorial design (a) and central composite design (b) for two variables. In red the levels of the factorial design and in blue the star points.

For the sake of clarity, the set of problem instances carried out according to the full factorial design and the central composite design are labelled T1 and T2, respectively.

Table 5.10 shows the factor levels for both the designs considering that the self radius (with respect to the first experimental campaign) is set in order to have the same value of the detector radius. The actual values per each tested combination are reported in Appendix (Table A2). Due to the deterministic generation of detectors based on an integer number of side divisions, the levels are not exactly scaled as in the theoretical design (-1.414, -1, 0, 1, 1.414) but as close as possible to such values. Five repetitions for different training data are carried out to have a rough assessment of the performance variability.

Table 5.10: Parameter levels.

Parameter	Low (L)	Middle Low (ML)	Intermediate (I)	Middle High (MH)	High (H)
Number of divisions	70	51	20	13	10
Self/Detector radius	0.0101	0.0139	0.0354	0.0544	0.0707
Censoring distance	r_{det}	–	$0.5(r_{self} + r_{det})$	–	$(r_{self} + r_{det})$

Table 5.11: Best combination and performance metrics per each test performed. The name of the test refers to the number of training samples and the repetitions. In some tests, more combinations had the same best score. Regarding the number of FP and FN it is worth remembering that the total real negative are 500 and real positive are 520. The six best overall performances (for 50, 150 and 300 training samples on T1 and T2) are highlighted in bold and grey.

T1							T2						
Test	dist	r_{det}	r_{self}	FP	FN	ROCAUC	Test	dist	r_{det}	r_{self}	FP	FN	ROCAUC
50-R1	H	L	H	25	94	0.885	50-R1	H	ML	MH	43	66	0.894
50-R2	H	L	H	18	81	0.904	50-R2	H	MH	MH	31	63	0.908
50-R3	H	L	H	24	96	0.884	50-R3	H	ML	MH	47	66	0.900
50-R4	H	L	H	19	95	0.900	50-R4	H	ML	MH	51	71	0.881
50-R5	H	L	H	13	111	0.880	50-R5	H	MH	MH	20	93	0.891
150-R1	H	L	I	37	62	0.903	150-R1	H	L	I	37	62	0.903
150-R2	I	I	H	46	64	0.892	150-R2	I	I	H	46	64	0.892
150-R3	L	H	L	33	71	0.899	150-R3	L	H	I	33	71	0.899
150-R3	L	H	I	33	71	0.899							
150-R3	L	H	H	33	71	0.899							
150-R3	I	H	H	33	71	0.899							
150-R4	H	L	I	37	66	0.900	150-R4	H	MH	ML	47	53	0.902
150-R5	H	I	I	22	84	0.897	150-R5	H	MH	ML	46	50	0.906
300-R1	I	L	H	25	85	0.893	300-R1	H	MH	ML	31	72	0.900
300-R2	H	L	I	11	83	0.909	300-R2	H	MH	ML	23	66	0.914
300-R3	H	I	L	27	72	0.904	300-R3	H	I	L	27	72	0.904
300-R4	I	L	H	18	80	0.905	300-R4	H	L	I	5	96	0.903
300-R5	H	L	I	16	77	0.910	300-R5	H	MH	ML	28	51	0.923

The data used for the training stage are randomly selected from the baseline, testing the influence of the number of data by comparing 50, 150 and 300 training samples. The data for the validating stage consists of 500 health-state data and 520 outliers (40 per each interval of 2°C) randomly generated.

Table 5.11 and Table 5.12 report the best combination given by each test with the corresponding performance score.

Table 5.12: Same of Table 5.11 but with explicit values.

T1				T2			
Test	dist	r_{det}	r_{self}	Test	dist	r_{det}	r_{self}
50-R1	0.0808	0.0101	0.0707	50-R1	0.0683	0.0139	0.0544
50-R2	0.0808	0.0101	0.0707	50-R2	0.1088	0.0544	0.0544
50-R3	0.0808	0.0101	0.0707	50-R3	0.0683	0.0139	0.0544
50-R4	0.0808	0.0101	0.0707	50-R4	0.0683	0.0139	0.0544
50-R5	0.0808	0.0101	0.0707	50-R5	0.1088	0.0544	0.0544
150-R1	0.0455	0.0101	0.0354	150-R1	0.0455	0.0101	0.0354
150-R2	0.0530	0.0354	0.0707	150-R2	0.0530	0.0354	0.0707
150-R3	0.0707	0.0707	0.0101	150-R3	0.0707	0.0707	0.0354
150-R3	0.0707	0.0707	0.0354				
150-R3	0.0707	0.0707	0.0707				
150-R3	0.0707	0.0707	0.0707				
150-R4	0.0455	0.0101	0.0354	150-R4	0.0683	0.0544	0.0139
150-R5	0.0707	0.0354	0.0354	150-R5	0.0683	0.0544	0.0139
300-R1	0.0404	0.0101	0.0707	300-R1	0.0683	0.0544	0.0139
300-R2	0.0455	0.0101	0.0354	300-R2	0.0683	0.0544	0.0139
300-R3	0.0455	0.0354	0.0101	300-R3	0.0455	0.0354	0.0101
300-R4	0.0404	0.0101	0.0707	300-R4	0.0455	0.0101	0.0354
300-R5	0.0455	0.0101	0.0354	300-R5	0.0683	0.0544	0.0139

Based on the available data, a response surface model is defined per each problem instance and each censoring distance. Such a multiple regression model allows to predict the behaviour of a dependent variable, y , based on a set of independent variables. In this study, the dependent variable is the ROCAUC, whereas the independent variables are the detector radius and the self radius. Thus, the model reads as follows:

$$y_i = \beta_0 + \beta_1 x_{i1}^2 + \beta_2 x_{i2}^2 + \beta_3 x_{i1} + \beta_4 x_{i2} + \beta_5 x_{i1} x_{i2} + \varepsilon_i \quad (\text{Eq. 5.43})$$

Considering that the censoring distance depends on the radius values, it is not considered as an independent variable, but different models for the different distance levels are formulated. The coefficients of the models and the plots of the surfaces are reported in Appendix (Table A2). In the Appendix are reported also the surface plots (Figure A1 and Figure A2). The maximisers of each surface function are presented in Table 5.13. For each instance, the best combination of censoring distance and radii, according to the prediction, is tested resorting to the same dataset used for the parameter setting. Due to the need of an integer division of the space side, the detector radius cannot be exactly equal to the maximiser of the response surface, however the difference in the predicted AUC, with the maximiser and with the tested value is negligible. In the table the values of the radius used in the test is reported. Comparing the predicted and the actual AUC values, four times out of six, the surface value overestimates

the actual one; the maximiser combinations outperform the ones provided by the design in three cases out of six (T1.1, T1.3 and T2.2); still, the difference between the performances is always less than 1%, except for instance T1.1.

Table 5.13: Maximisers and maximum of the response surfaces. In bold the best performance per each instance.

<i>Instance</i>	<i>dist level</i>	r_{det}	<i>Tested r_{det}</i>	r_{self}	<i>dist</i>	<i>Pred. AUC</i>	<i>Actual AUC</i>
T1.1	L	0.070		0.013	0.070	0.791	
T1.1	I	0.0388		0.0700	0.054	0.820	
T1.1	H	0.0412	0.0416	0.0532	0.094	0.917	0.902
T1.2	L	0.067		0.0118	0.067	0.873	
T1.2	I	0.01	0.0101	0.0688	0.039	0.891	0.899
T1.2	H	0.01		0.0418	0.052	0.872	
T1.3	L	0.0614		0.01	0.061	0.886	
T1.3	I	0.01	0.0101	0.0639	0.037	0.907	0.914
T1.3	H	0.01		0.01	0.020	0.899	
T2.1	L	0.07		0.046	0.070	0.782	
T2.1	I	0.07		0.07	0.070	0.828	
T2.1	H	0.0346	0.0354	0.0508	0.085	0.915	0.892
T2.2	L	0.0538		0.07	0.054	0.895	
T2.2	I	0.0292		0.0610	0.045	0.901	
T2.2	H	0.0328	0.0337	0.0328	0.066	0.916	0.911
T2.3	L	0.0514		0.07	0.051	0.923	
T2.3	I	0.0316	0.0321	0.0526	0.042	0.929	0.914
T2.3	H	0.0310		0.01	0.041	0.916	

The outcomes of the previous analyses are largely confirmed, moreover it can be noted that:

- The CCD provides, in 13 cases out of 15, the same or a better performance score, whereas the use of the surface response does not provide a significant improvement to the performance;
- In a few cases, and only for 150 or 300 samples, thus for a good coverage of the self space, the best ratio between detector and self radii is inverted, namely the best combination has larger detector than self radius. In these cases, the censoring distance is larger than or equal to the detector radius. Comparing the FFD and the CCD for the same instances in which this inversion happens (e.g. 300-R1 for T2, see Table 5.11 and Table 5.12), the performance is improved as the classifier with larger detector radius produces more FP but the reduction in percentage of FN is larger than the increment in percentage of the FP.

5.3. Segmental masonry arch

5.3.1. Description of case study and experiments

The second case study concerns a small-scale segmental masonry arch (Figure 5.19), built and tested in the structural laboratory of the Institute of Bio-Sustainability of the University of Minho [292], [293]. The arch specimen consists of four rows of 39 brick units ($100 \times 75 \times 50 \text{ mm}^3$) assembled with staggered lime mortar joints. It has a nominal span of 1900 mm, a springing angle of 40° , a nominal net rise of 430 mm and a 75 mm radial thickness. The structure is symmetrically loaded with two lime bags of 25 kg each (corresponding to about 50% of the arch weight), in order to simulate the presence of backfill material typical of masonry arch bridges. Two concrete abutments support the system: one fixed to the floor by bolts (left support) and the other set on a simple system with lateral roulette wheels (right support) featuring a movable springing along the horizontal direction.

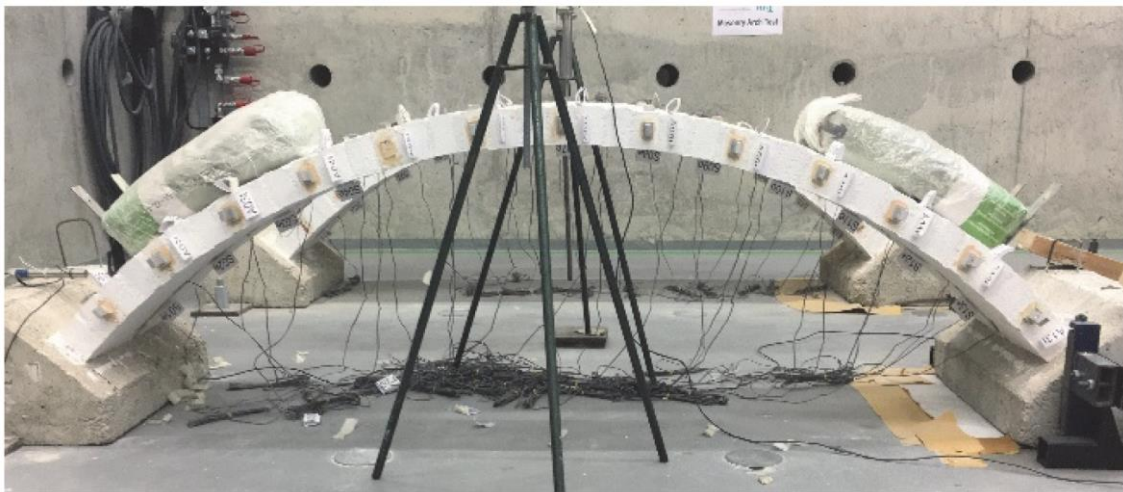


Figure 5.19: Configuration of the masonry arch.

The main goal of the experiment is the investigation of the effects of support settlements on the dynamic behaviour of masonry arch bridges. Therefore, the small-scale arch is progressively damaged by applying uniform increasing displacements at one support. After each displacement stage, dynamic tests are performed to measure the dynamic response of the system and follow its stiffness degradation due to the occurrence of structural damage. For each incremental step, the structural response is measured in terms of strains and accelerations. Regarding the former, twenty-six linear strain gauges (series PL-120-11-5L) are used both at the intrados and extrados of the arch middle line (locations Sxx in Figure 5.20). The latter are recorded through eight accelerometers (model PCB 393B12, 0.15 to 1000 Hz frequency range, 10000 mV/g sensitivity, $8 \mu\text{g}$ resolution), of which four are kept fixed, whereas the other four are

moved according to 12 consecutive set-ups. This process allows to collect records from 26 points evenly spaced along the front and back edges of the arch both in normal and tangential direction, for a total number of 52 acceleration responses (locations Axx in Figure 5.20). Such a dense sensor distribution enables to have a high accuracy of the mode shape estimates. For the acceleration records, an acquisition window of at least 2000 times the fundamental period of the arch is required. Hence, considering also that the frequency of interest for the structure fell below 200 Hz, the signals are sampled at 400 Hz for a minimum duration of 180 s, resulting in 72.000 data points per channel. In built heritage preservation, vibration monitoring is a widespread and well-established methodology, therefore, in the present work, we focused only on the acceleration records.

During the measurements, other mechanical equipment and machines are working in the laboratory, inducing a non-negligible background noise. Therefore, two acquisitions per each set-up are repeated considering the following inputs: (1) ambient excitations and (2) random finger tapping. The latter produces a fairly high signal-to-noise ratio, without violating the assumption of stationary Gaussian white noise input on which operational modal analysis relies. In this second case, shorter acquisitions are acceptable, so the minimum duration of 60 s (24.000 data points per channel) is set.

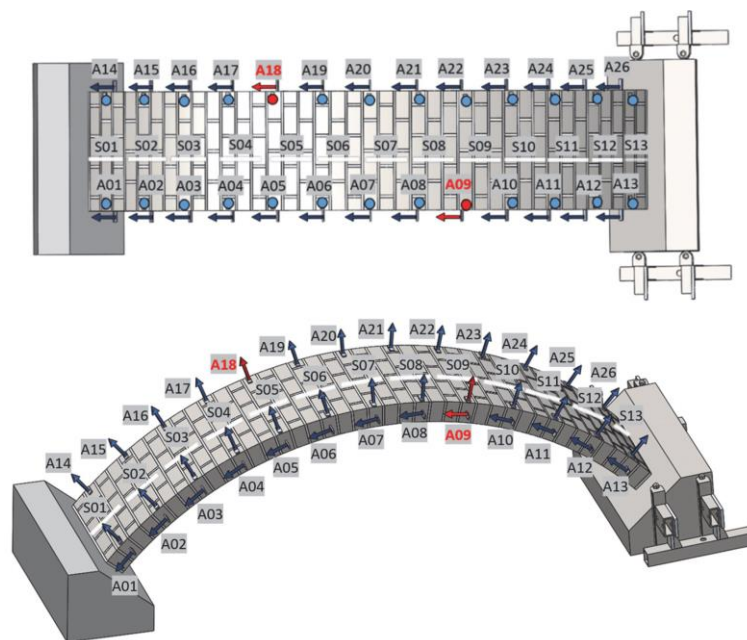


Figure 5.20: Location of the measurement points for the strains (S) and the accelerations (A).

The displacement-controlled test is carried out in five consecutive steps, generating five unrecoverable Damage Scenarios (DSs). Outward movements in horizontal direction are imposed to the right support with a hydraulic jack, whose displacement is simultaneously measured by a Linear Variable Displacement

Transducer (LVDT). The consecutive scenarios, with the relative horizontal displacements, are reported in Table 5.14. After five steps, the arch reaches 3 mm of settlement at the crown. Then, the displacement is further increased until the collapse.

Table 5.14: Support settlements in the consecutive damage scenarios.

Stage	Name	Acronym	Settlement [mm]
0	Reference Scenario	D0	—
1	Damage scenario 1	DS1	0.4
2	Damage scenario 2	DS2	0.5
3	Damage scenario 3	DS3	0.6
4	Damage scenario 4	DS4	0.7
5	Damage scenario 5	DS5	0.8
6	Collapse		

After the first step, three cracks appear at the mortar-brick interfaces (Figure 5.21): *c1* at the intrados, in the left region of the keystone; *c2* at the extrados of the left springing (fixed support), and *c3* at the extrados of the right springing (moving support). The outset of cracks induces a considerable drop of the structural stiffness which is clearly reflected by the downshift of the main natural frequencies of the arch (particularly for modes 1, 4 and 5). The crack pattern observed at the end of the five stages is presented in Figure 5.22.

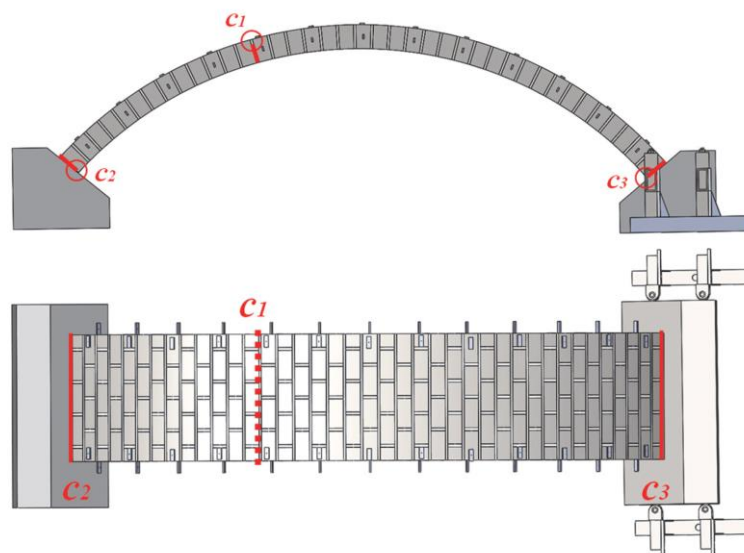


Figure 5.21: Location of the three hinges during the test.

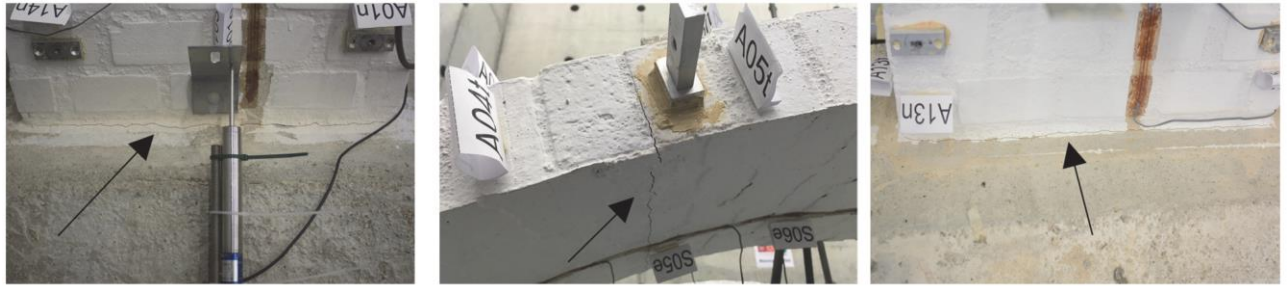


Figure 5.22: Crack pattern induced by the horizontal settlement.

The acquired data are processed through the software ARTeMIS Modal [294], using two well-known estimators: the Enhanced Frequency Domain Decomposition (EFDD) and the Stochastic Subspace Identification with Extended Unweighted Principal Component (SSI-UPCX). Both the estimators allow to identify, for the D0 configuration, 6 modes in the range 30-135 Hz with damping ratios below 5%.

In Table 5.15 the results of the SSI-UPCX estimator are reported in terms of averaged frequency values for all the meaningful vibration modes over the different scenarios.

Table 5.15: Eigenfrequencies variation over progressive damage scenarios.

Scenario	Frequencies [Hz]				
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
D0	30.06	50.95	59.44	95.23	120.62
D1	26.31	50.28	59.04	80.47	113.16
D2	23.4	48.86	58.14	75.75	111.44
D3	21.44	45.14	58.27	74.97	110.6
D4	20.20	44.05	57.77	74.39	110.98
D5	19.16	43.19	57.54	74.09	110.86
$\Delta f(D0-D5)$ [%]	-36.26	-15.23	-3.20	-22.20	-8.09

For the aim of the present study, the frequency downshift after the first induced displacement is already too significant to require an automatic detection. Thus, in order to test the algorithm in unfavourable conditions, the classification is performed by analysing the values of the second and third natural frequencies, whose relative variation between scenarios is the smallest. Moreover, only the undamaged and the first two damaged conditions are considered. During the test, twelve set-ups have been considered with 4 accelerometers that stayed always in the same position and 4 that moved. This provides a set of 96 acquisitions for each scenario: 48 from the moving accelerometers and 48 from the reference ones. Considering the two input sources (e.g. ambient and finger tapping) the dataset amounts to 192 points. However, the distributions of the samples under the two typologies of input (Figure 5.23) differ to such an extent that they cannot be used together. This means that, in a real application, the input might affect the classification and new samples must be collected under the same type of action than the ones used

for the training. In order to simulate the most challenging scenario, the ambient vibration data are considered for the damage detection. This dataset, as shown in Figure 5.23, has a larger variance, likely due to the worse signal-to-noise ratio that might complicate the classification. Moreover, operating on built heritage, ambient vibration tests are more suitable than forced vibration, therefore testing the methodology against these data is important in view of real-world applications.

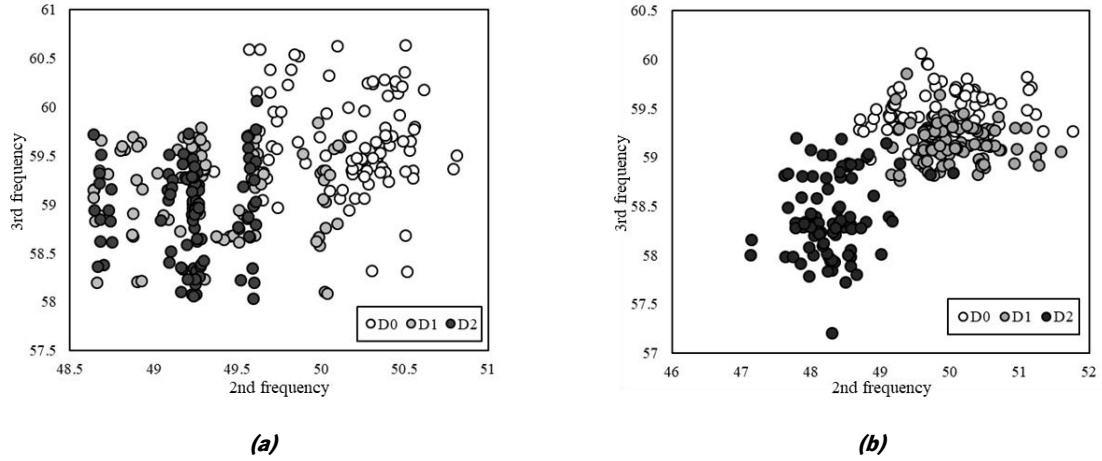


Figure 5.23: Samples under ambient vibration (a) and finger tapping (b).

The absence of bias in this sampling is qualitatively investigated by plotting the outcomes. The distribution does not look significantly affected by the sensor location (Figure 5.24) or by the set-up (Figure 5.25). For these reasons, in the following analysis, we consider a consistent dataset composed of 96 values extracted through peak-picking directly from the power spectral density of each acquisition under ambient vibration in each damage condition. Most of the results discussed hereafter have been presented in [295] and published in [296].

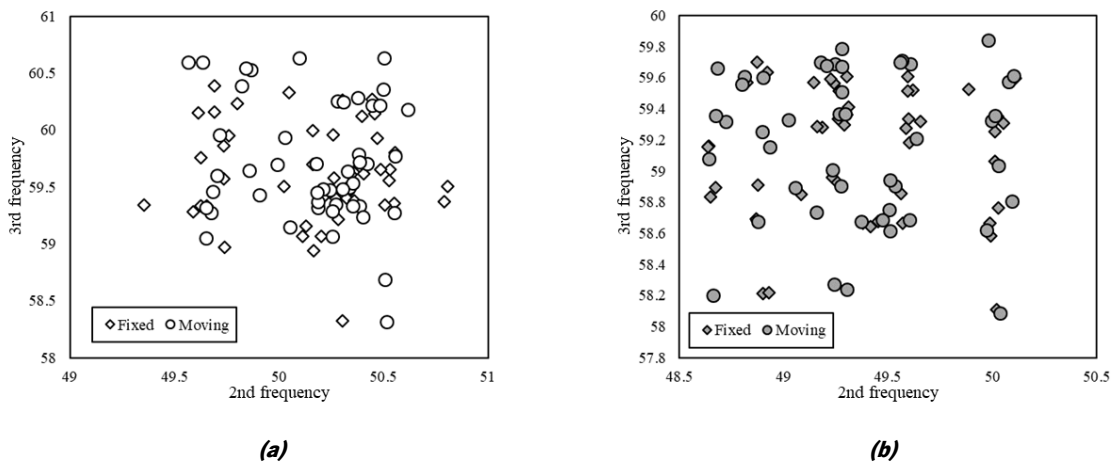


Figure 5.24: Samples under ambient vibration divided in fixed and moving sensors: (a) D0 condition and (b) D1 condition.

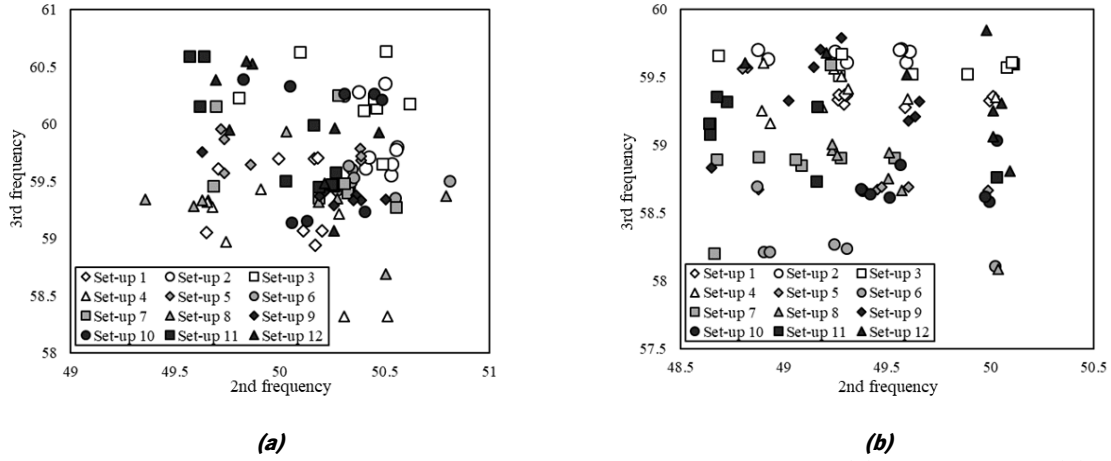


Figure 5.25: Samples under ambient vibration divided in the different set-ups from (a) D0 condition and (b) D1 condition.

5.3.2. Description of the experimental analysis

In the following, 4 sets of analyses are discussed: (1) validation of the DGNSA over real field-testing data; (2) comparison of the results with other classification algorithms; (3) simulation of a classifier training with reduced knowledge (e.g. no real damaged samples available); (4) simulation of online monitoring.

The dataset, including self and nonself real samples, is first used to perform a validation of the algorithm and a parametric analysis assessing the influence of the following parameters:

- Self radius r_{self} ;
- Detector radius r_{det} ;
- Censoring distance formulation.

The adaptation rate η_0 is set equal to 0.005 and the maturity age t to 30.

Considering that only 96 self samples are available, three scenarios are tested:

- T1: 40 self samples are available for the training and 56 self + 56 nonself for the validation;
- T2: 50 self samples are available for the training and 46 self + 56 nonself for the validation;
- T3: 60 self samples are available for the training and 36 self + 56 nonself for the validation.

Based on the knowledge gained through the analysis of data from the church of the Monastery of Jerónimos, at least 40 samples are considered necessary to perform a good training. The tested levels for each parameter are the same reported in Table 5.10 and the actual values in each combination are reported in Appendix (Table A3). To account for the randomness, five repetitions are carried out for each number of samples. The upper and lower bounds of the unitary space are recalculated based on each training sample and the elements of the validating sets which fall outside the bounds are automatically labelled as false positive and true positive for the self and nonself points, respectively. The performance metrics are estimated including these elements.

For the sake of brevity, among the comparison tests, only the ones based on the instance T2 are reported. Due to the limited size of the dataset, this instance is the one that offers the best balance between a representative training set and a sufficient validation set. The analysis aims to compare the quality of the solution provided by the DGNSA with two different algorithms suitable for one-class classification.

The five repetitions of instance T2 are also used for the offline detector generation considering a reduced level of knowledge, namely assuming that only a small set of real samples from the healthy structure is available. In this case, two sets of healthy and unhealthy data are artificially simulated. The best classifier is then tested against real self and nonself data and its performance is compared with the results of the analysis of instance T2. Three approaches are considered:

- T2.1: Fixed self radius with simulated self data used for validation;
- T2.2: Fixed self radius with simulated self data used for training;
- T2.3: Variable self radius according to a Gaussian method with simulated self data used for validation.

Finally, the best classifier identified offline by analysing instance T2 is implemented to the online monitoring of the data in order to test the damage detection strategy based on the positive counters described in chapter 3, section 3.2.2. In this case, the monitored data are considered as ordered following acquisitions, assuming that the system is initially in healthy condition, then subjected to damage scenario D1 and finally to damage scenario D2.

5.3.3. Algorithm validation

The validation is performed considering both a three-level full factorial design (FFD) and a central composite design (CCD) repeated for the three formulations of the censoring distance. The combinations of factors and the levels are the same reported in Table 4.4 of Chapter 4 for the FFD and Table 5.10 of the present chapter for the CCD. A complete table with the tested values is reported in the Appendix (Table A3). The first set of experiments employs the FFD to define the best classifier for instances T1, T2 and T3, taking into account real measured samples for both training and validation. The AUC is calculated only considering the points used for validation. Even though scenarios D1 and D2 are the evolution of the same phenomenon, in general, different classifiers might be more or less effective against different damage scenarios. Hence, in the analyses and in the tables hereafter, the classifiers are validated and tested on a single damage scenario per time. Due to this, the confusion matrix has different TPs and FNs counts for D1 and D2 whereas, being the self samples the same in both the cases, TNs and FPs counts are coincident. Moreover, the performance of each classifier is averaged over the five repetitions and the

best overall classifier is the one that, on average, outperforms the others (even though it might result less accurate than others in a specific repetition).

Table 5.16 reports the results of the best combination of factor levels after the validation (e.g. self and detector's radii equal to 0.0707, censoring distance equal to their sum), for instance T1. For this instance, this combination is the most accurate in each repetition except for repetition 2.

Table 5.16: Results in the 5 repetitions of the best combinations on the average in instance T1.

Repetition	Training		Validating				Metrics		
	TN	FP	TN	FP	TP1	FN1	TNR	TPR	AUC
1	40	0	55	1	40	16	0.982	0.714	0.848
2	40	0	53	3	36	20	0.946	0.643	0.795
3	40	0	53	3	39	17	0.946	0.696	0.821
4	40	0	47	9	46	10	0.839	0.821	0.830
5	40	0	53	3	36	20	0.946	0.643	0.795
AVG	40	0	52.2	3.8	39.4	16.6	0.932	0.703	0.819
Repetition	TN	FP	TN	FP	TP2	FN2	TNR	TPR	AUC
1	40	0	55	1	48	8	0.982	0.857	0.920
2	40	0	53	3	43	13	0.946	0.768	0.857
3	40	0	53	3	46	10	0.946	0.821	0.884
4	40	0	47	9	51	5	0.839	0.911	0.875
5	40	0	53	3	42	14	0.946	0.750	0.848
AVG	40	0	52.2	3.8	46	10	0.932	0.821	0.877

Table 5.17: Results in the 5 repetitions of the best combinations on the average in instance T2.

Repetition	Training		Validating				Metrics		
	TN	FP	TN	FP	TP1	FN1	TNR	TPR	AUC
1	50	0	45	1	40	16	0.978	0.714	0.846
2	50	0	45	1	36	20	0.978	0.643	0.811
3	50	0	43	3	39	17	0.935	0.696	0.816
4	50	0	39	7	46	10	0.848	0.821	0.835
5	50	0	44	2	35	21	0.957	0.625	0.791
AVG	50	0	43.2	2.8	39.2	16.8	0.939	0.700	0.820
Repetition	TN	FP	TN	FP	TP2	FN2	TNR	TPR	AUC
1	50	0	45	1	48	8	0.978	0.857	0.918
2	50	0	45	1	43	13	0.978	0.768	0.873
3	50	0	43	3	46	10	0.935	0.821	0.878
4	50	0	39	7	50	6	0.848	0.893	0.870
5	50	0	44	2	42	14	0.957	0.750	0.853
AVG	50	0	43.2	2.8	45.8	10.2	0.939	0.818	0.878

For the case of T2 (Table 5.17) and T3 (Table 5.18), the outcome is slightly more complicated, as the best combination considering the averaged performance and the best combination within each repetition

tends to be different. The best combination reported in the table is, for T2, the same as T1. For T3, the combinations are different for the 2 scenarios: for D1, intermediate detector's radius, high self radius and censoring distance (e.g. detector's radius equal to 0.0354, self radius equal to 0.0707, censoring distance equal to their sum); for D2, the same as T1 (e.g. self and detector's radii equal to 0.0707, censoring distance equal to their sum). It is worth mentioning that an increasing number of samples for the training, in the investigated range (e.g. 40-60), does not significantly improve the performance.

Table 5.18: Results in the 5 repetitions of the best combinations on the average in instance T3.

Repetition	Training		Validating				Metrics		
	TN	FP	TN	FP	TP1	FN1	TNR	TPR	AUC
1	60	0	33	3	40	16	0.917	0.714	0.815
2	58	2	33	3	43	13	0.917	0.768	0.842
3	60	0	35	1	39	17	0.972	0.696	0.834
4	60	0	28	8	50	6	0.778	0.893	0.835
5	60	0	28	8	44	12	0.778	0.786	0.782
AVG	59.6	0.4	31.4	4.6	43.2	12.8	0.872	0.771	0.822
Repetition	TN	FP	TN	FP	TP2	FN2	TNR	TPR	AUC
1	60	0	36	0	45	11	1.000	0.804	0.902
2	60	0	36	0	43	13	1.000	0.768	0.884
3	60	0	36	0	39	17	1.000	0.696	0.848
4	60	0	32	4	49	7	0.889	0.875	0.882
5	60	0	34	2	42	14	0.944	0.750	0.847
AVG	60	0	34.8	1.2	43.6	12.4	0.967	0.779	0.873

Aiming at comparing different methodologies for the parameter setting, the CCD is also applied to instance T2 and the results are shown in Table 5.19. The best combination for D1 is obtained from intermediate detector's radius, high self radius and censoring distance (e.g. detector's radius equal to 0.0354, self radius equal to 0.0707, censoring distance equal to their sum), whereas, for D2, it is given by middle high detector's and self radius and high censoring distance (e.g. self and detector's radii equal to 0.0544, censoring distance equal to their sum). The best combination of the CCD is outperformed by the best one of the FFD, since the latter (where all the factors are set to high values) is not tested by the CCD. The lower performance is due to a slightly more aggressive behaviour, as more positive points are correctly classified (more TPs and less FNs) but there is a larger relative increment of misclassified negative points (FPs). It is worth reminding that the adopted performance metric, namely the ROC AUC, considers equally the two types of errors. In structural monitoring, however, FPs might be considered less severe than FNs. The definition of a weighted performance metric, to include this preference, falls out of the scope of the present work.

Table 5.19: Results in the 5 repetitions of the best combinations on the average in instance T2 considering a CCD.

Repetition	Training		Validating				Metrics		
	TN	FP	TN	FP	TP1	FN1	TNR	TPR	AUC
1	50	0	40	6	10	16	0.870	0.714	0.792
2	50	0	45	1	22	18	0.978	0.679	0.828
3	50	0	43	3	12	14	0.935	0.750	0.842
4	50	0	38	7	9	11	0.844	0.804	0.824
5	50	0	43	3	23	19	0.935	0.661	0.798
AVG	50	0	41.8	4	15.2	15.6	0.912	0.721	0.817
Repetition	TN	FP	TN	FP	TP2	FN2	TNR	TPR	AUC
1	50	0	39	7	6	50	0.848	0.893	0.870
2	50	0	42	4	10	46	0.913	0.821	0.867
3	50	0	43	3	8	48	0.935	0.857	0.896
4	50	0	38	7	7	49	0.844	0.875	0.860
5	50	0	41	5	9	47	0.891	0.839	0.865
AVG	50	0	40.6	5.2	8	48	0.886	0.857	0.872

Based on such results, it is clear that the performance improves with increasing censoring distance and increasing self and detector radii.

The above outcome apparently disagrees with the results of the previous case study. Though, it must be highlighted that the good performance provided by a large detector radius is likely due to the sparsity of the training samples in the feature space. Detectors with small radius occupy the space within the training samples instead of only surrounding them (e.g. see Figure 5.26). The combination with low level of the detector radius and high level of the self radius and of the censoring distance was the best one in most of the tests performed in the case study of the church of Jerónimos. In this second case study, the combination still has an acceptable performance, however it presents a behaviour more aggressive than the best combination overall (compare Figure 5.27 and Figure 5.28). In conclusion, the importance of the parameter setting and the need of an improved strategy for the approximation of the self space are confirmed, which also consider the case-specific needs (e.g. sparsity of the samples, characteristics of their distribution in the feature space, etc.). Nonetheless, the use of a small detector radius and a large self radius is recommended when no further tests on the dataset are performed.

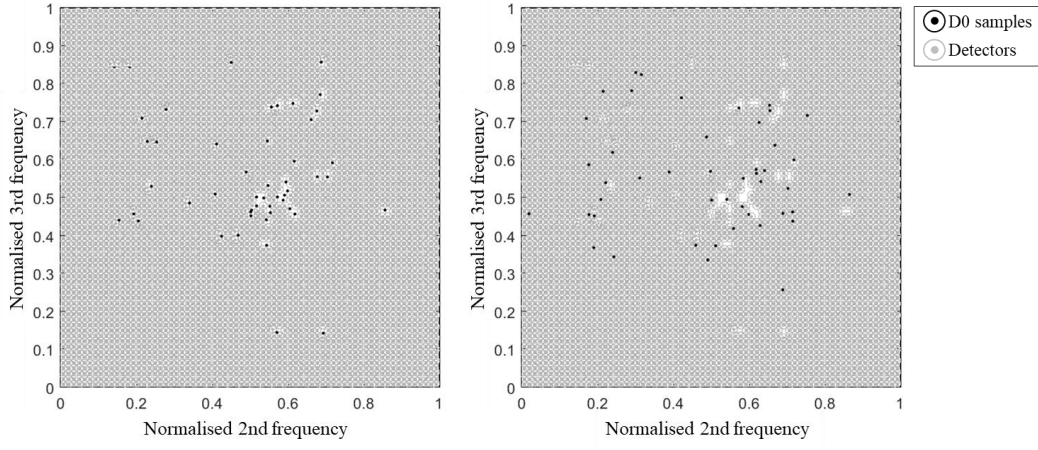


Figure 5.26: Feature space for T2 repetition 1 - combinations with low values for all the factors: (a) Training self samples against detectors; (b) Validating self samples against detectors.

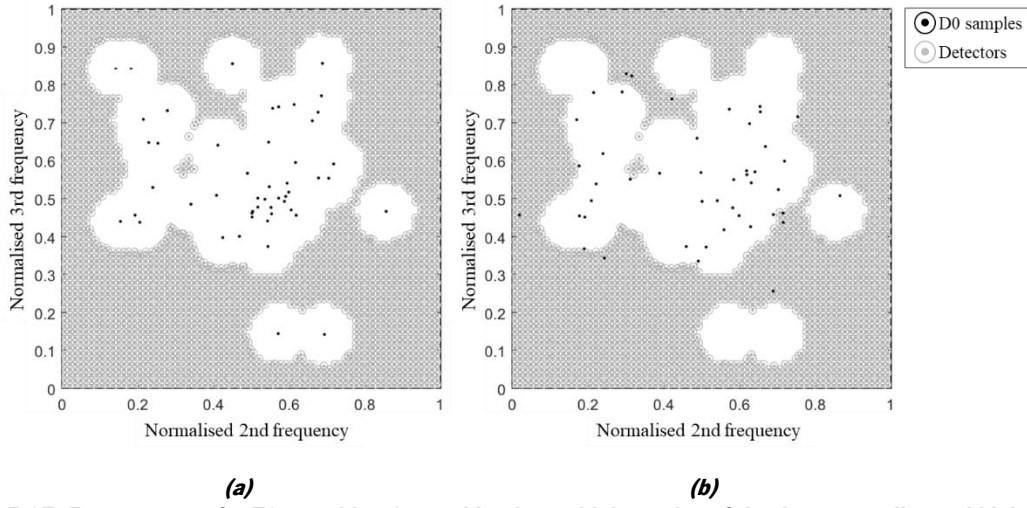


Figure 5.27: Feature space for T2 repetition 1 - combinations with low value of the detector radius and high values of the self radius and of the censoring distance: (a) Training self samples against detectors; (b) Validating self samples against detectors.

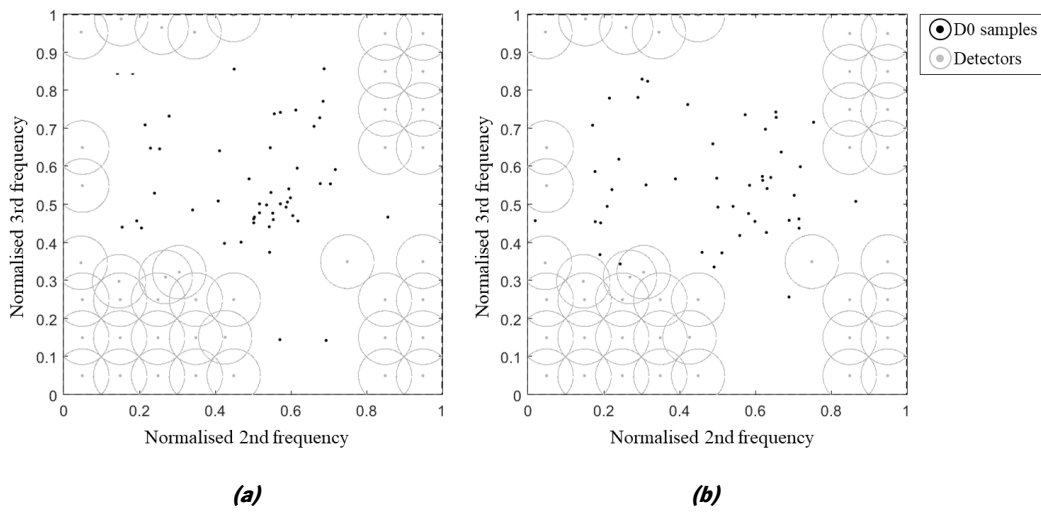


Figure 5.28: Feature space for T2 repetition 1 - combinations with high values for all the factors: (a) Training self samples against detectors; (b) Validating self samples against detectors.

5.3.4. Algorithm comparison

In order to assess the performance of the DGNSA algorithm, the five repetitions of instance T2 are compared with two well-known algorithms for classification. The first one is a probabilistic classifier based on the Mahalanobis squared distance [297], [298]. Mahalanobis squared distance is a measure of discordance for multivariate data whose formulation is given by the following expression:

$$D_{\zeta} = (\{x\}_{\zeta} - \{\bar{x}\})^T [S]^{-1} (\{x\}_{\zeta} - \{\bar{x}\}) \quad (\text{Eq. 5.44})$$

where $\{x\}_{\zeta}$ is the potential outlier, $\{\bar{x}\}$ is the mean of a reference sample set (e.g. the training data) and $[S]$ is the covariance matrix. For the classification, in the present work, the threshold is defined by calculating the inclusive Mahalanobis squared distance of each sample in the training set and assuming the maximum distance as detection threshold. Given any sample to be classified, its Mahalanobis squared distance from the training set samples is measured. The sample is classified as nonself if the distance is larger than the threshold, and self otherwise. Table 5.20 reports the outcome of the classification for the five repetitions of instance T2. It is clear that the classifier has a more tolerant behaviour, since it reduces the number of FPs, but significantly increases the number of FNs.

Table 5.20: Confusion matrix for the five repetitions of instance T2 and average confusion matrix (ANS and AS indicate the actual nonself and self, respectively; LNS and LS designate labelled nonself and self, respectively).

	R1		R2		R3		R4		R5		AVG	
	LNS	LS	LNS	LS	LNS	LS	LNS	LS	LNS	LS	LNS	LS
ANS	45	1	46	0	44	2	41	5	46	0	44.4	1.6
AS	40	16	29	27	39	17	40	16	29	27	35.4	20.6

A more effective methodology for binary classification is the Support Vector Machine (SVM). A thorough description of this algorithm can be found in [299], [300]. The SVM aims at the definition of the best hyperplane in the feature space which separates the points belonging to the two classes, namely the hyperplane with the largest margin between them. The points of each class which are the closest to the hyperplane are called support vectors. Whenever the classes cannot be separated by a simple hyperplane, the SVM methodology can be still applied but it requires the definition of a map for the points, from the original space to a linear space where the hyperplane can be defined. The mapping is a nonlinear transformation based on the theory of reproducing kernels. Many functions belonging to different classes are suitable for the transformation. In this study, a radial basis function (RBF) kernel is used and the SVM is applied to one-class classification.

In order to achieve a fair comparison, the algorithm is trained on the same normalised dataset without any other manipulation of the data using the same hardware and software supports. The need for a fair

comparison has been stressed by many authors in the field of Computer Science [118], [255], [257]. Indeed, the comparison of an algorithm under development with an SVM toolbox already implemented in MATLAB might be unfair for several reasons, such as differences in the programmer factors and unequal attention to the used coding techniques. However, the comparison experiment was designed mainly to check the quality of the solution provided by the present methodology without focusing on other aspects as computational effort. Moreover, double checking a problem instance through other methodologies allows to confirm that a good performance is not due to a too simplistic case study. A preliminary pilot study performed by the authors proved the SVM output to be largely affected by the value of the Kernel Scale Parameter. In order to study the behaviour of the algorithm for different values of the Kernel scale, a sensitivity analysis is performed by performing 201 runs for each repetition of instance T2 (five in total), varying the parameter value within the range $[10^{-3}, 10^3]$. To guarantee a similar effort in the parameter setting of the NSA, the simple FFD is not considered enough, therefore a sensitivity analysis is carried out by considering the intermediate value of the detector radius, the largest formulation for the censoring distance and 96 possible values for the self radius in the range $[0.005, 0.1]$.

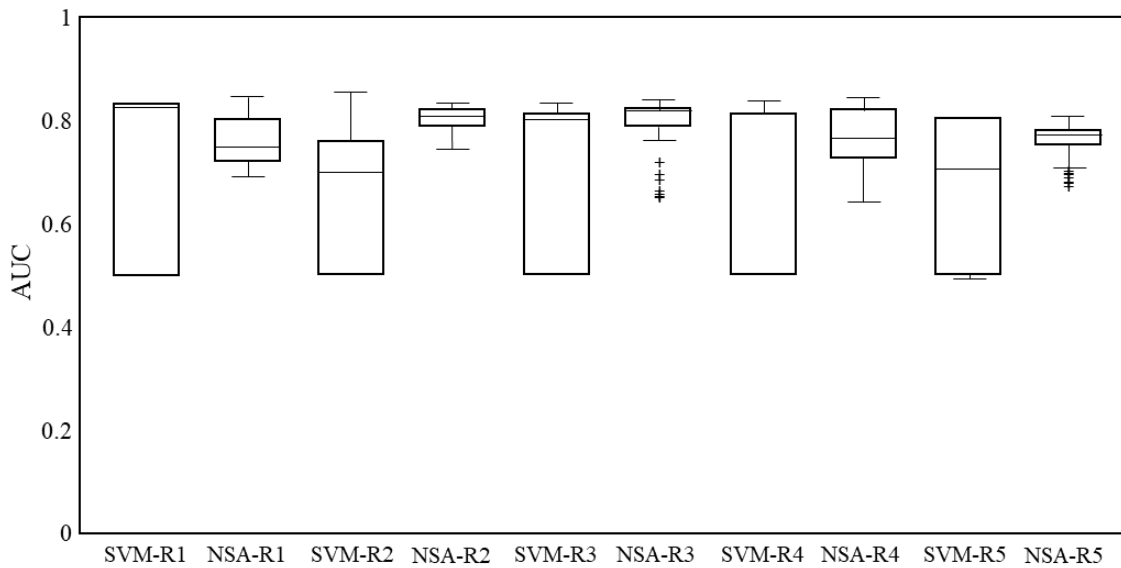


Figure 5.29: Boxplot of the AUC for the five repetitions of instance T2 and damage scenario D1, analysed through SVM and NSA with different parameters values.

The results are presented through the boxplot of Figure 5.29 with reference to the AUC for the first damage scenario (D1). It is worth noting that the DGNSA provides the highest AUC in four repetitions out of five and shows, in general, a much slighter variation in terms of performance. This might be due to the larger range of values that the Kernel scale can assume. The lowest AUC of the SVM is always equal to

0.5 which is the AUC of a random classifier, however the median of the performance is very close to maximum value and in two repetitions out of five it is larger than the median of the NSA's AUC. As for the NSA, the median tends to be more central within the distribution of the AUC, meaning that the parameter setting affects relatively more the performance. In conclusion, the quality of the solution provided by the NSA is considered acceptable, being comparable - if not even better - than the solution provided by a well-developed and validated version of the SVM.

5.3.5. Reduced knowledge during training

In the following test, the real nonself data are used only to examine the classifier, after the validation which is performed by using artificial data. Namely, the NSA algorithm is used to generate classifiers with different parameter settings by training on a small set of real samples from the healthy structure and validating on two sets of healthy and unhealthy artificial data. The best classifier is then tested against real nonself data and its performance is compared with the results of the previous section. Again, for simplicity, only the results of the instance T2 are discussed. Three approaches are tested:

- T2.1: Fixed self radius with simulated self used for validation;
- T2.2: Fixed self radius with simulated self used for training;
- T2.3: Variable self radius according to Gaussian method (radius value for the max probability density sample equal to 0.1) with simulated self used for validation.

The 5 training samples sets are used to define the feature space and to simulate self and nonself data. In cases 1 and 3, the same sets are employed to train the algorithm, whereas in case 2 the simulated self samples are used for training and the real self samples are used for validation. In cases 1 and 2 the same levels of the algorithm parameters are tested, whereas in case 3 the self radius does not need to be tuned, thus only nine tests are performed considering three formulations of the censoring distance and three values of the detector radius.

Table 5.21, Table 5.22 and Table 5.23 report the results of the best combination in each repetition. On average, the best classifier in each instance is:

- T2.1: intermediate self and detector radii, high censoring distance;
- T2.2: low detector radius, intermediate self radius and censoring distance;
- T2.3: intermediate detector radius and high censoring distance.

Table 5.21: Results in the five repetitions of the best combinations on the average in instance T2, case 1.

Repetition	Training		Validating					Metrics	
	TN	FP	TN	FP	TP	FN	TNR	TPR	AUC
1	50	0	753	181	906	543	0.806	0.625	0.716
2	50	0	779	159	1176	270	0.830	0.813	0.822
3	50	0	800	155	677	451	0.838	0.600	0.719
4	50	0	631	119	638	86	0.841	0.881	0.861
5	50	0	732	198	1138	346	0.787	0.767	0.777
AVG	50	0	739	162.4	907	339.2	0.820	0.737	0.779

Repetition	Testing D1					Metrics		
	TN	FP	TP	FN	TNR	TPR	AUC	
1	30	16	41	15	0.652	0.732	0.692	
2	39	7	44	12	0.848	0.786	0.817	
3	37	9	43	13	0.804	0.768	0.786	
4	26	20	49	7	0.565	0.875	0.720	
5	32	14	47	9	0.696	0.839	0.767	
AVG	32.8	13.2	44.8	11.2	0.713	0.800	0.757	

Repetition	Testing D2					Metrics		
	TN	FP	TP	FN	TNR	TPR	AUC	
1	30	16	53	3	0.652	0.946	0.799	
2	39	7	52	4	0.848	0.929	0.888	
3	37	9	51	5	0.804	0.911	0.858	
4	26	20	53	3	0.565	0.946	0.756	
5	32	14	52	4	0.696	0.929	0.812	
AVG	32.8	13.2	52.2	3.8	0.713	0.932	0.823	

After the training and the validation, the classifiers are tested over the same dataset used for the validation of T2. Therefore, the results of the testing stage for T2.1, T2.2. and T2.3 are compared with the results of the validation stage of T2. In the three cases, the best combination achieved in the validation stage is not the one that would provide the best performance with real nonself data. This means that a classifier validated over artificially simulated nonself data cannot perform against a specific damage scenario as well as if it was validated over real nonself data belonging to the same scenario. Nonetheless, the performance is promising with a TPR that is on average close to or larger than 80% against D1 and 90% against D2. In T2.1 and T2.3, the best combination over real nonself data would be the one with high values for the factors, whereas in T2.2 it is the one having high detector radius and censoring distance, with low self radius for D1 and intermediate self radius for D2. This last result is consistent with the use of a much larger dataset for the training that prevents the need for a large self radius to increase the self space coverage (Figure 5.30). Compared with the results over the instance T2, a deterioration of the performance becomes evident due to an aggressive behaviour with a small improvement of the TPR but a large reduction of the TNR. The best performance is provided by T2.2. The classifier of T2.3 is the most aggressive. Nonetheless, the performance is comparable to the other cases. This turns the methodology adopted for the self approximation worthy of attention. In fact, the plot in Figure 5.31 proves that the

same self coverage could be achieved with less samples. This further reduces the number of acquisitions needed upstream to train the algorithm, increasing its cost-effectiveness.

Table 5.22: Results in the five repetitions of the best combinations on the average in instance T2, case 2.

Repetition	Training		Validating				Metrics		
	TN	FP	TN	FP	TP	FN	TNR	TPR	AUC
1	934	0	46	4	1005	444	0.920	0.694	0.807
2	938	0	45	5	1116	330	0.900	0.772	0.836
3	955	0	45	5	728	400	0.900	0.645	0.773
4	750	0	45	5	670	54	0.900	0.925	0.913
5	930	0	45	5	1193	291	0.900	0.804	0.852
AVG	901.4	0	45.2	4.8	942.4	303.8	0.904	0.768	0.836

Repetition	Testing D1				Metrics		
	TN	FP	TP	FN	TNR	TPR	AUC
1	39	7	40	16	0.848	0.714	0.781
2	41	5	46	10	0.891	0.821	0.856
3	38	8	37	19	0.826	0.661	0.743
4	25	21	49	7	0.543	0.875	0.709
5	35	11	45	11	0.761	0.804	0.782
AVG	35.6	10.4	43.4	12.6	0.774	0.775	0.774

Repetition	Testing D2				Metrics		
	TN	FP	TP	FN	TNR	TPR	AUC
1	39	7	50	6	0.848	0.893	0.870
2	41	5	51	5	0.891	0.911	0.901
3	38	8	48	8	0.826	0.857	0.842
4	25	21	56	0	0.543	1.000	0.727
5	35	11	45	11	0.761	0.804	0.782
AVG	35.6	10.4	50	6	0.774	0.893	0.849

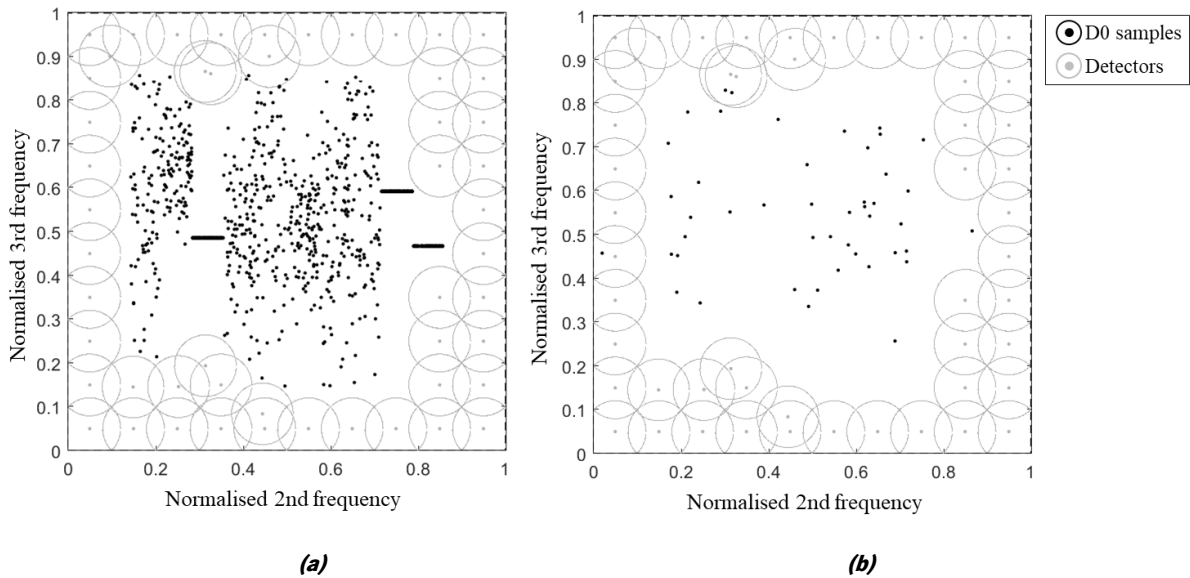


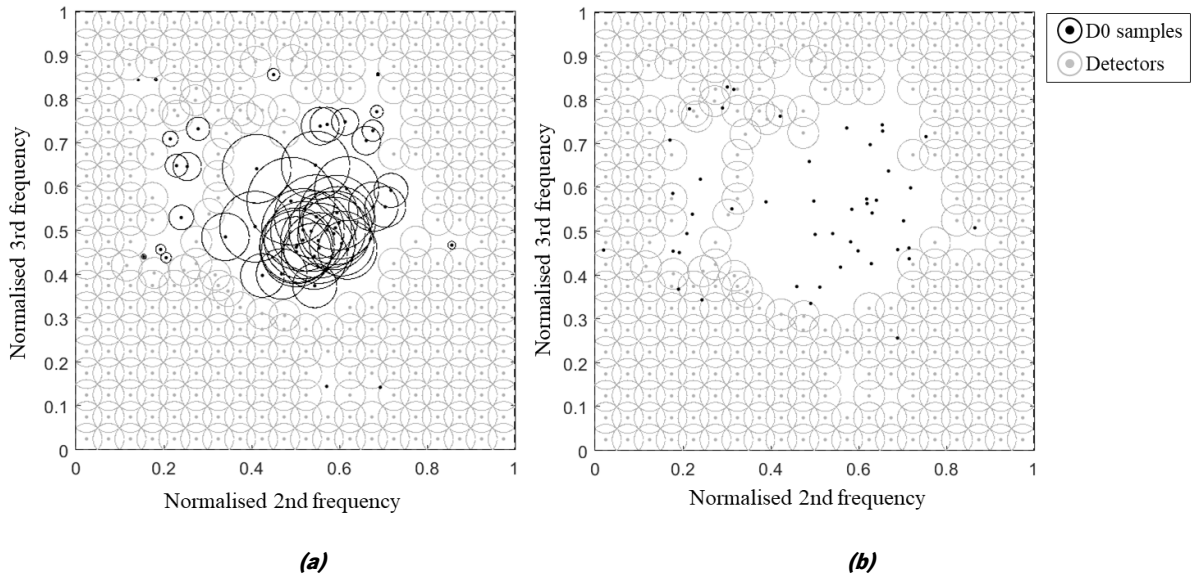
Figure 5.30: Feature space for T2.2 repetition 1, best combination: (a) Training self samples against detectors; (b) Validating self samples against detectors.

Table 5.23: Results in the five repetitions of the best combinations on the average in instance T2, case 3.

Repetition	Training		Validating					Metrics	
	TN	FP	TN	FP	TP	FN	TNR	TPR	AUC
1	50	0	728	206	1048	401	0.779	0.723	0.751
2	50	0	664	274	1247	199	0.708	0.862	0.785
3	50	0	734	221	789	339	0.769	0.699	0.734
4	50	0	555	195	719	5	0.740	0.993	0.867
5	50	0	658	272	1283	201	0.708	0.865	0.786
AVG	50	0	667.8	233.6	1017.2	229	0.741	0.828	0.785

Repetition	Testing D1					Metrics		
	TN	FP	TP	FN	TNR	TPR	AUC	
1	30	16	46	10	0.652	0.821	0.737	
2	36	10	45	11	0.783	0.804	0.793	
3	37	9	47	9	0.804	0.839	0.822	
4	22	24	52	4	0.478	0.929	0.703	
5	34	12	45	11	0.739	0.804	0.771	
AVG	31.8	14.2	47	9	0.691	0.839	0.765	

Repetition	Testing D2					Metrics		
	TN	FP	TP	FN	TNR	TPR	AUC	
1	30	16	55	1	0.652	0.982	0.817	
2	36	10	53	3	0.783	0.946	0.865	
3	37	9	54	2	0.804	0.964	0.884	
4	22	24	56	0	0.478	1.000	0.739	
5	34	12	53	3	0.739	0.946	0.843	
AVG	31.8	14.2	54.2	1.8	0.691	0.968	0.830	

**Figure 5.31: Feature space for T2.3 repetition 1, best combination: (a) Training self samples with variable radius against detectors; (b) Validating self samples against detectors.**

5.3.6. Monitoring simulation

The five repetitions of instance T2 are employed to test the improved strategy based on positive and negative counters. This strategy balances the readiness of the warning with the risk of false alarms, since it informs that the system might be damaged not after the first positive label but after a given number of

consecutive positives. The strategy, in fact, exploits the fact that the damage is not reversible in terms of system response, thus, once the system is altered, the features change consistently from the distribution due to the health state to a new one. It is assumed that the samples used for the validation form a set of consecutive acquisitions from the structure. The damage outbreak (e.g. scenario D1) happens after 46 consecutive acquisitions in healthy conditions. After 56 additional acquisitions the damage evolves to the scenario D2. Figure 5.32 reports the sequences of acquisition along time and shows the damage outbreak and evolution.

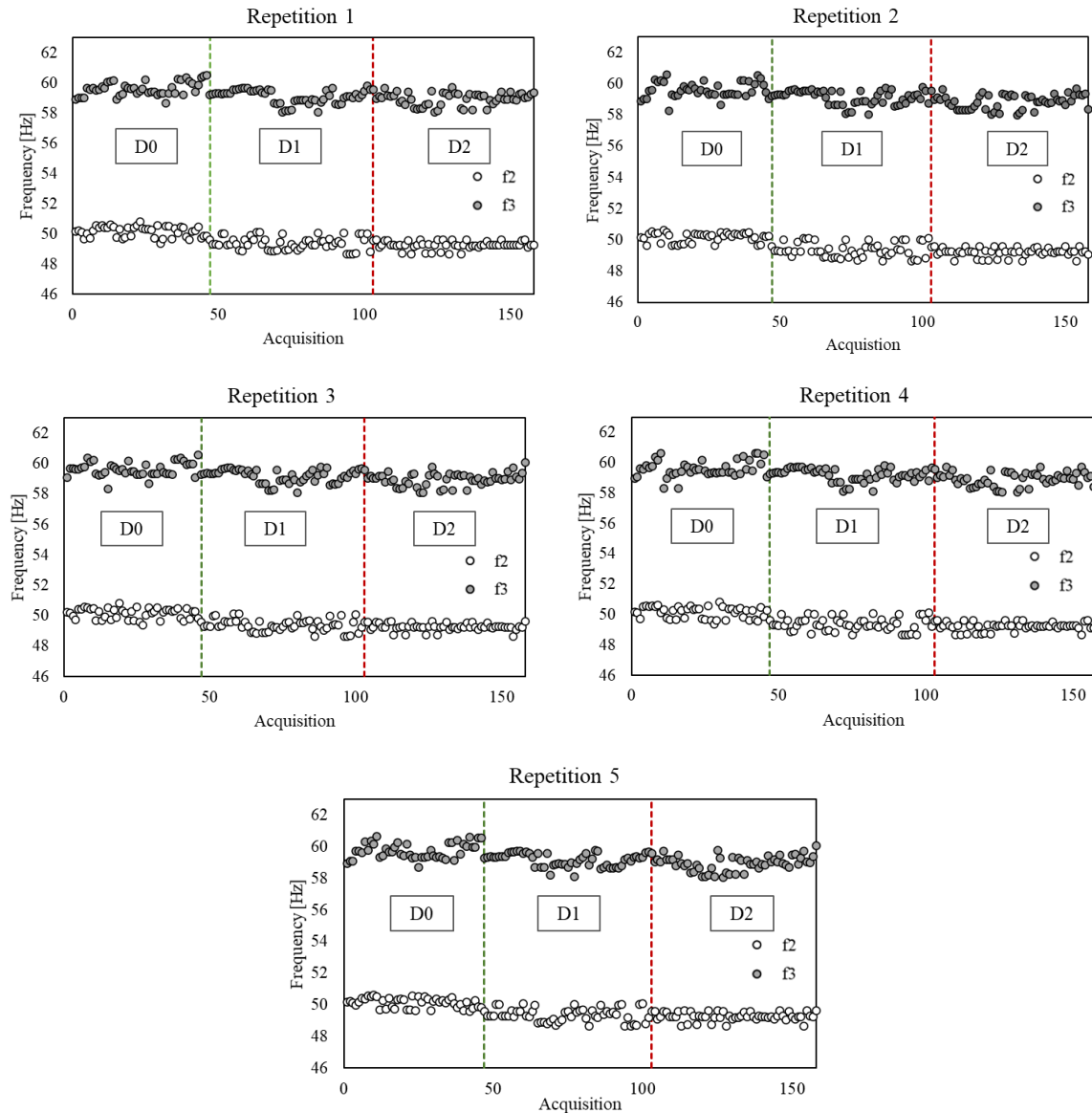


Figure 5.32: Sequence of the acquisitions for the five repetitions of instance T2, the vertical lines identify the shift to a following damage scenario.

The analysis is carried out only on the scenarios D0 and D1 (Figure 5.33). The success score of the best classifier in each of the repetitions is reported in the picture, namely 1 if the classifier assigns the correct

label and 0 if it does not. The counter strategy needs to set a positive and a negative threshold and a reset value. Each new consecutive positive label increases the positive counter by one, until the positive threshold, when the alarm is fired. However, if after a series of positive labels, the classifier produces negative labels, the positive counter is frozen and, eventually, reset if the number of consecutive negative labels reach the reset threshold value. For instance, considering the result of Figure 5.33, some self elements are classified as nonself, the maximum number of consecutive positive labels produced over the self samples in the five simulations is 2 (e.g. the positive counter is equal to 2), assuming that the positive threshold was set to a value larger than 2 and that the reset value was equal to 1 (namely the counter is reset after a single negative label) no false alarm would be sent.

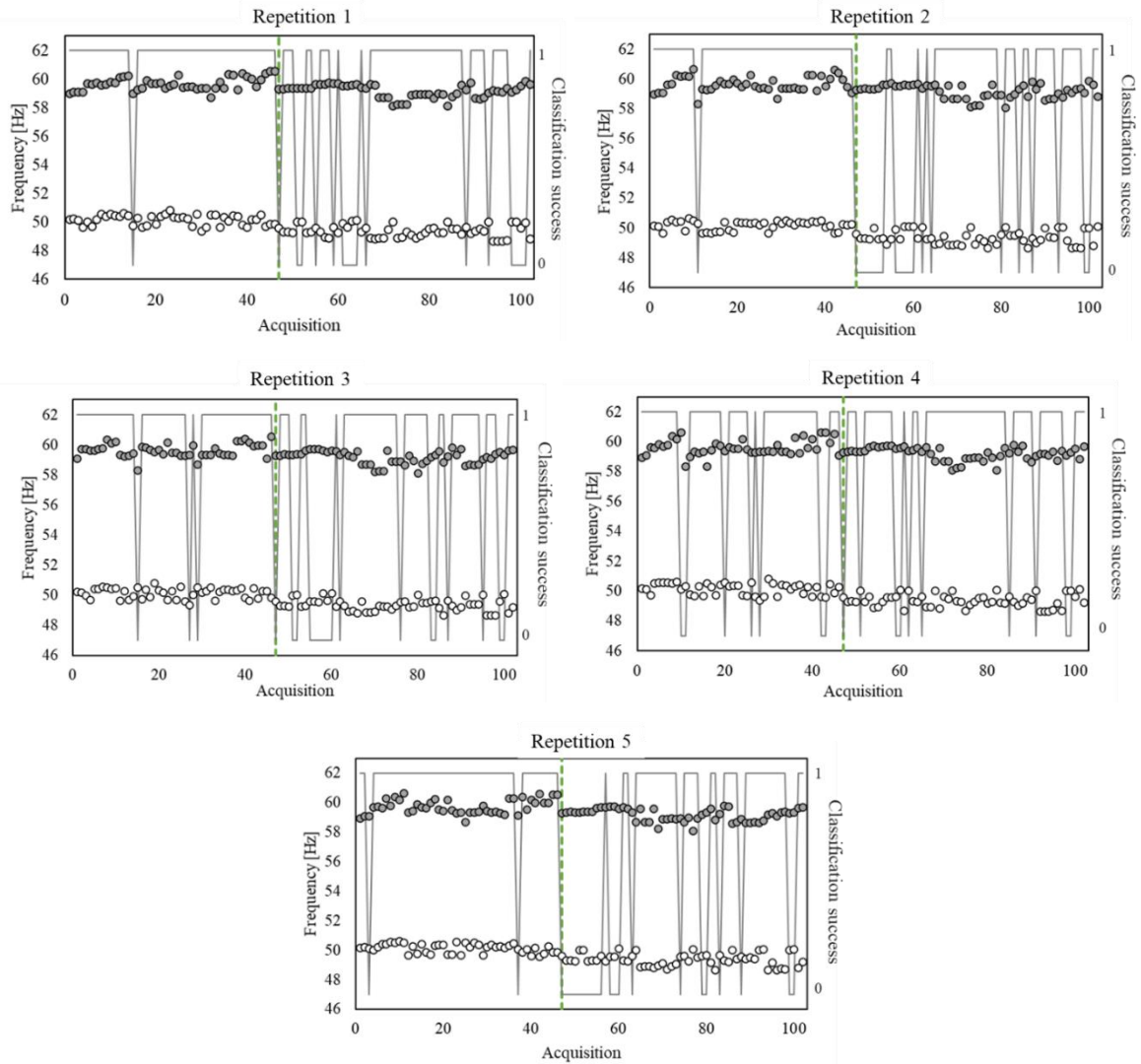


Figure 5.33: Classification success for scenarios D0 and D1. A score equal to 1 means correct labelling, while 0 means incorrect labelling.

Focusing on the damaged system, it is worth analysing the readiness of the classifier considering different combinations of positive threshold and reset value. Figure 5.34 shows the growth of the positive counter for different reset values (namely 1, 2 and 3). As the reset value increases within the same number of acquisitions, the maximum value of the positive counter increases significantly and the number of acquisitions required to achieve at least a given value of the positive counter decreases. Thus, a larger reset value increases the readiness for the same positive threshold.

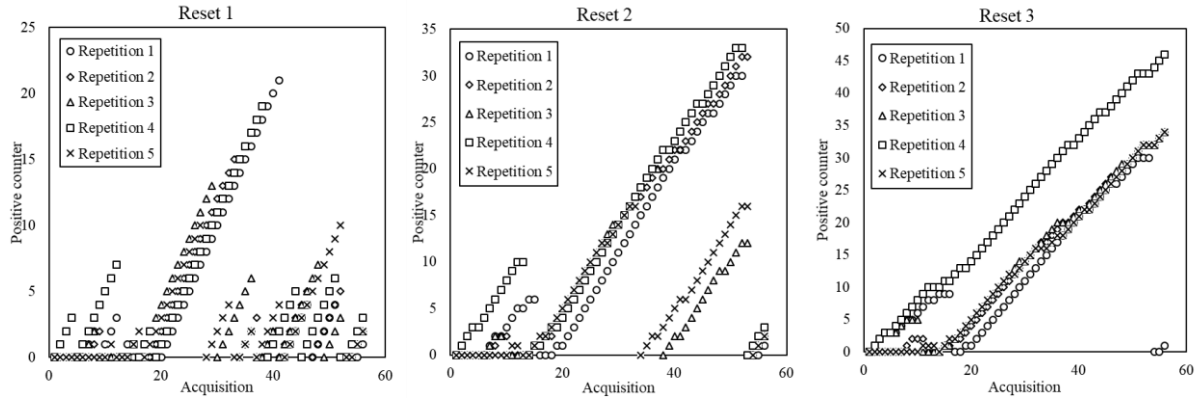


Figure 5.34: Evolution of the positive counter in the five repetitions of the instance considering different reset values.

This result is confirmed by the output reported in Table 5.24, where the number of acquisitions needed to fire the alarm, after the damage outbreak, is presented for different positive thresholds P_T and reset values R_v . The readiness can be explained in sight of the True Positive Rate. On average, given a single unseen damaged point in instance T2, the classifier correctly labels it 70% of the times (30% failure rate), see Table 5.17. Thus, the probability of firing the alarm, within a given number of acquisitions, can be forecast through tree diagrams, as the two shown in Figure 5.35. There is the 70% of probability that the first acquisition after the system changed condition from D0 to D1 is classified as nonself (LNS in the figure). Assuming a reset value of 2 and a positive threshold of 2 (as in Figure 5.35a), the alarm is fired after the second acquisition only if both the acquisitions are labelled as nonself. This has a probability of 49%. The alarm is fired after the third acquisition whenever the sequence of labels is $\{LS, LNS, LNS\}$ or $\{LNS, LS, LNS\}$. If the reset value was set to 1, though, in the second case the positive counter would have been reset to 0 after the second acquisition and the alarm would not have been fired after the third. The tree diagrams, as the two shown in Figure 5.35, help proving that, for the same threshold, a larger reset value reduces the number of acquisitions required. However, the two values should be set in order to avoid a false positive alarm (FPA in Table 5.24). As reported in the examples of Table 5.24, in fact, a small threshold with a large reset value increases the chance that the alarm is fired when the system is

still in healthy conditions. For a reset of at least 2 and a threshold of 2 this happens in two repetitions out of five.

Table 5.24: Number of acquisitions needed to achieve a given positive threshold P_T for different reset values (FPA stands for False Positive Alarm).

P_T	$R_v = 1$						$R_v = 2$						$R_v = 3$					
	R1	R2	R3	R4	R5	FPA	R1	R2	R3	R4	R5	FPA	R1	R2	R3	R4	R5	FPA
2	3	9	3	3	16	R4	3	9	3	3	16	R3-R4	3	9	3	3	16	R3-R4
3	4	21	4	4	20	No	4	19	4	4	18	No	4	19	4	4	18	No
4	24	22	20	9	21	No	11	20	19	6	19	No	7	20	7	6	19	No
5	25	23	21	10	22	No	12	21	20	7	20	No	8	21	8	7	20	No
6	26	24	22	11	23	No	14	22	21	8	21	No	10	22	21	8	21	No
7	27	25	23	12	24	No	26	23	22	9	22	No	11	23	22	9	22	No
8	28	26	24	27	25	No	27	24	23	10	23	No	12	24	23	10	23	No
9	29	27	25	28	26	No	28	25	24	11	24	No	14	25	24	11	24	No
10	30	28	26	29	27	No	29	26	25	12	25	No	29	26	25	12	25	No

Finally, in Table 5.25, the probability of firing the alarm within a given number of acquisitions for different combinations of threshold and reset values is reported. This probability is equal to the sum of the probabilities of firing the alarm in the previous acquisitions and in the last one: for instance, the probability of firing the alarm within 4 acquisitions is equal to the sum of the probabilities after acquisition 1, 2, 3 and 4. A similar estimation can be performed for the system in healthy condition, then the positive threshold and the reset value can be defined in order to guarantee a given readiness with a given probability of false positive alarm.

Table 5.25: Probability of firing an alarm within at a given acquisition between 1 and 6 for different combinations of threshold and reset values.

Acquisition	1	2	3	4	5	6
$P_T = 2 R_v = 2$	0%	49%	78.40%	87.22%	92.95%	96.53%
$P_T = 2 R_v = 3$	0%	49%	78.40%	91.63%	95.60%	97.72%
$P_T = 3 R_v = 2$	0%	0%	34.30%	65.17%	77.52%	84.00%
$P_T = 3 R_v = 3$	0%	0%	34.30%	65.17%	83.69%	91.10%

5.3.7. Conclusion

In the present chapter, most of the features described in chapter 3, for the definition of an integrated damage detection strategy based on a negative selection algorithm, were tested against the data produced by field-testing. The first analysed case, namely the church of the Monastery of Jerónimos, in Lisbon, offered a large dataset but in a single system condition. This allowed to focus only on the performance of the algorithm in terms of the approximation of the self space. The second case, namely a segmental masonry arch tested in laboratorial conditions, despite providing a much smaller dataset from controlled conditions - which might be non-representative of a large scale application - allowed to analyse data from real damage scenarios. The influence of the algorithm qualitative and quantitative parameters on the performance was deeply assessed, confirming in great part the outcome of the analysis against the numerical data of chapter 4. It was evident that the performance is affected by the combination of self radius, detector radius and censoring distance, and the best setting depends on the nature of the training and validating sets. In more detail, when the self space was properly approximated, a small detector radius and a large self radius with a censoring distance equal to the sum of these two radii provided on the average a higher performance. However, when the training set was large and dense, the self radius could be reduced, thereby achieving a better approximation of the boundary between self and nonself, hence a better performance. On the opposite, small and sparse training set required a larger censoring distance and a larger detector radius to avoid the generation of detectors within the self space. Validating set was essential to balance true positive and true negative rates, to help selecting the most accurate classifier. Considering the influence of the training and validating sets characteristics on the performance of the classifier, two strategies were developed and applied. The first aimed at assessing the quality of the training set belonging to the reference state, based on two metrics (e.g. the area within the enveloping boundary of the samples and the incremental value of the area for each new sample). The resulting samples were likely to be representative of the feature's self space. The second aimed at artificially generating nonself data based on the distribution of the self training set. Although the best classifier after validation through artificial data was suboptimal, its performance, in the tests, was deemed sufficient to successfully detect an abnormal behaviour.

The parameter setting proved to be essential for the performance of the methodology, especially when the training set was small. Increasing the size of the training set, the variability in the performance tended slightly to flatten. It was not possible to identify a design of the experiment strategy able to outperform the others. When the best values of the factors fell within the tested range (as for the Monastery of Jerónimos), the central composite design exploring more values could provide a better result; however,

when the best values were pushed to the bounds of the range, the full factorial design resulted better as it tested those combinations. The methodology here developed allowed to support the damage detection by defining a boundary between the features belonging to the normal behaviour and the abnormal ones, adapting to complex nonconvex self spaces and requiring a very limited knowledge. The tests performed in the present chapter demonstrated that, even when few samples are available, despite belonging only to the baseline condition, the algorithm could be successfully trained and the resulting classifier provides an acceptable performance, comparable - if not superior - to other well-known methods.

CHAPTER 6

Conclusions and Future Lines of Research

6.1. Summary and conclusions

This thesis addressed the problem of early-stage damage detection in Civil Engineering structures by applying nature-inspired computing techniques, in particular a new version of Negative Selection Algorithm (NSA). The work was divided in four phases: (1) review of damage detection strategies within the field of Structural Health Monitoring and accurate analysis of the applications related to nature-inspired algorithms, focusing on NSA; (2) definition of the methodology for damage detection based on a new version of the NSA with deterministic generation; (3) application and validation through numerical instances, one replicating a frame building and one representing a well-known benchmark bridge; (4) application and validation through field-testing case studies, a segmental masonry arch tested in laboratory and the church of the monastery of Jerónimos, in Lisbon. In the following sections, the main conclusions from the thesis are summarized, together with remarks for future works.

6.1.1. Damage detection methodology based on Negative Selection Algorithm

Damage detection is the first step of a wider damage identification process that can be summarized, using Rytter's hierarchy, according to five main goals of increasing complexity: (1) Detection of existence; (2) Localisation; (3) Classification of the type; (4) Quantification of the extent; (5) Prognosis.

The damage detection is basically a measure of the discordance between sets of damage-sensitive features, representing the system in known conditions, and values of the same features in the actual state. An estimate of the disagreement is then compared to a threshold set to distinguish satisfactory behaviours and faults. Mathematically, this can be formulated as a classification problem in a "limited" supervised learning way, also called anomaly detection or one-class classification. The training data, namely the feature representing the structural behaviour that are known before starting the detection, belong only to one of the two classes, that is the reference state. Recently, a branch of Computer Science called Natural Computing has been concerned with the development of innovative numerical techniques that are inspired by natural biological processes and phenomena. Such techniques demonstrated to be extremely efficient when applied to complex problems. Among them, Negative Selection Algorithms (NSAs) have been successfully applied to one-class classification. NSAs are a family of algorithms based on a minimal common framework, initially developed in the 90s of the last century and later improved with additional contributions. In the present study, a novel version of an NSA was developed and used as the main core of a complete bio-inspired damage detection strategy. In order to tackle the shortcomings and limitations of existing procedures that emerged from the literature review, this new version aimed at optimising the detector set, for low-dimensional feature spaces, by introducing a deterministic generation

strategy. Being a one-class classification algorithm, in the beginning of the process it has to be trained on a set of samples belonging to the reference (undamaged) state. However, since different algorithm parameter settings are likely to provide different classification results, in the developed methodology the initial generation is repeated according to a proper Design of Experiment (DOE) strategy and the performance of the classifiers is analysed on a validating dataset where also nonself elements are needed. Since in many practical situations, real data are not available for the nonself conditions, a strategy for an effective artificial generation was implemented. The classifiers are ultimately compared to identify the best one according to one or more performance metrics. Several indicators were listed and analysed for this purpose. During the monitoring stage, the selected classifier analyses any new sample provided by the monitoring system. Two possible improvements for the methodology were presented. One aims at preventing the risk of false alarms by using counters of the consecutive self or nonself classifications. The other aims at higher order classification either using the distinct sensitivity of more classifiers in parallel to different damage scenarios or considering the signatures as damage-sensitive features. The methodology proved to be suitable for the continuous online and real-time assessment of the system safety, and to be independent of the typology of structure as well as of the analysed damage-sensitive features.

6.1.2. Numerical case studies

The damage detection methodology was validated by analysing its most relevant features over two artificial datasets: a concrete frame building and the I-40 bridge. Numerical case studies allow to simulate damage scenarios and monitoring processes through FE models of fictitious or real structures, to produce large artificial datasets and to control all the relevant parameters that are needed to identify the weaknesses and the flaws of the methodology.

In particular, the first case study, a concrete frame building, presents a very low complexity, thus it was used as a preliminary testbed for the approach. In this case, the fluctuation of structural response under changing environmental and operational conditions was simulated by a stochastic variation of the live load, and a deterministic relationship between the Young's modulus and the temperature. Six different damage scenarios were considered by reducing the stiffness of a pillar at one floor. The tests allowed to identify the algorithm parameters with larger influence on the performance and their range of variation. The second case study has a slightly higher complexity. The I-40 bridge is a well-known benchmark in the field of SHM, being a bridge that was fully monitored under induced damage scenarios before demolition. The same incremental damage was reproduced, in the numerical model, by gradually cutting the middle

span of the north plate girder. Moreover, the output data used for the damage detection were obtained by simulating a structural monitoring through accelerometers, allowing to test the methodology in a more realistic situation. Unlike the other case studies presented in the thesis, the signatures (power spectral densities or Fast Fourier transforms of the recorded accelerations) were directly considered as damage-sensitive features.

The numerical analyses allowed to draw some conclusions also regarding the damage sensitivity of different features produced by a dynamic monitoring, although this aspect falls outside the scope of the present work. In particular, in the first case study, several pairwise combinations of natural frequencies, with or without considering the ambient temperature, were tested. Promising results were obtained by performing the classification on pairs of natural frequencies or pairs of differences between natural frequencies, which proved to be quite interesting since it does not require the monitoring of other factors (e.g. the temperature). In general, classifiers trained in different 2-D feature spaces showed very distinct performance against the simulated scenarios. This not only confirmed that the identification of the quantities of interest is a fundamental step of the monitoring design, but it also suggested that such a distinct sensitivity can be exploited to improve the chances of correct detection against many possible unknown damage scenarios and to extend, when possible, the classification to higher orders, including damage localisation and quantification. The higher order classification was also achieved by following a completely different approach in the second case study, where each measurement is represented by a set of values, namely the signature amplitude per each frequency line. The anomaly score distribution of each damage scenario was used as reference for higher order classification.

6.1.3. Field-test case studies

Finally, the methodology was applied and validated through two real case studies. Like in the numerical instances, the most relevant components of the damage detection strategy were tested separately, comparing different suitable approaches. First, the dataset from the long-term monitoring of the church of the monastery of Jerónimos, in Lisbon, was analysed. This comprised the evolution of three natural frequencies over five years with the corresponding values of temperature and relative humidity. Such a large dataset was an incredibly valuable source for the tests, especially in terms of the approximation of the self space, although all the samples belonged to the system in its undamaged condition. During this period, in fact, there was no evidence of emerging behaviours likely linked to the occurrence of damage. The second case, namely a reduced-scale segmental masonry arch tested in the laboratory, did not have the complexity of the previous instance and provided a much smaller dataset. However, it allowed to

analyse data from real damage scenarios. Being a laboratory simulation, the acquisitions and the application of the damage happened in controlled operational and environmental conditions. This, on the one hand, could be non-representative of a real-world application, however, on the other hand, ensured a larger control on the source of uncertainties.

Based on these analyses, the influence of the qualitative and quantitative parameters on the algorithm performance was estimated, largely confirming the outcome of the analyses through numerical instances. The performance was clearly affected by the combination of self radius, detector radius and censoring distance, whose best setting depended on the nature of the training set. In more details, when the self space was properly approximated (e.g. the training set was not scarce and not sparse with respect to the self population), a small detector radius and a large self radius with a censoring distance equal to the sum of the two tended to provide a higher detection performance. With very large and dense training sets, the self radius could be reduced to improve the self space approximation in the boundary with the nonself space. On the opposite, small and sparse training sets, providing a poorer approximation of the actual self space, required a larger censoring distance and a larger detector radius to avoid the generation of detectors within the self space. It was demonstrated that the parameter setting strongly affects the performance, especially when the training set was small. The pre-defined range for the parameters, as well as the strategy used for setting, influenced the results. When increasing the size of the training set, the variability in the performance slightly tended to flatten.

6.2. Future lines of research

The methodology here developed allows to support the damage detection task by defining a boundary between the features belonging to the normal behaviour and those belonging to the abnormal ones, adapting to complex nonconvex self spaces and requiring a very limited knowledge. Indeed, several numerical strategies were implemented and tested to improve the accuracy of the classifier when only the reference condition is known and when the training set is sparse. A tailored methodology and some metrics were developed to optimise the number of acquisitions for a more cost-effective preliminary monitoring of the system. Finally, additional solutions were introduced to reduce the risk of false alarms and extend the classification to a higher order by supervise learning. The tests performed in the present thesis were promising and proved that the algorithm is suitable for online and real-time damage detection, with a comparable, if not superior, performance when assessed against other well-known methods. The objectives discussed in the introduction of the present thesis have been successfully achieved and the main issues emerged during the literature review have been tackled with computational techniques that,

in some cases, might be beneficial and suitable to other damage detection methodologies, even not NSA-based. However, some future lines of research are indicated hereafter both to address interesting aspects that emerged during the tests - and could not be analysed within this work - and to tackle other issues that were not originally in the scope of the thesis but would be essential to further improve the effectiveness of the methodology. The following topics are recommended.

Regarding the preliminary monitoring, the self approximation strategy used to optimise the number of acquisitions has been tested *a-posteriori* on the dataset from the church of Monastery of Jerónimos. The results are indeed promising, however, to prevent any biased conclusion, testing the methodology for future monitoring that are not performed yet is deemed necessary.

Future investigations should also aim to improve the algorithm validation and testing. Comparing different scaling algorithms, it emerged that some transformations led to the misclassification of samples with less probability of occurrence. In this case, the artificial nature of the dataset did not allow a confident generalisation of the results. Therefore, it is strongly recommended that such beneficial property be verified also for real-world datasets and, eventually, that a performance metric taking into account not only the correctness of the classification but also the probability of the mistaken samples be formulated. Since scaling is a common stage of many anomaly detection strategies, other damage identification approaches beyond the NSA would benefit from this line of research.

The methodology has not been fully casted yet into a probabilistic framework. In this regard, to improve the generalisation of the validation results, possible advanced solutions can be implemented and tested using enhanced sampling strategies to form training and validating sets (e.g. bootstrap), fusing the information provided by different metrics and considering them as non-deterministic.

A probabilistic approach might be beneficial not only during the validation but, more in general, during the monitoring itself as it might provide confidence intervals for the classification or its likelihood. To this end, casting the classification into a Bayesian framework is an appealing solution that would require more efforts and research.

The main goal of the present thesis was to develop an early-stage damage detection methodology and test its effectiveness. At this stage, only a simplified comparison with alternative well-known techniques has been carried out to verify the goodness and reliability of the performance. Once a more refined version of the algorithm is available, a well-designed comprehensive numerical testing campaign for additional comparisons is strongly recommended.

The use of counters and the substructuring of the damage detection into more classifiers in parallel showed promising results in terms of increased reliability of the detection, reduced risk of false alarms

and potential to extend the methodology to higher order classification. These characteristics are very appealing in view of real-world applications, however they have been tested mainly considering artificial data, over a limited number of feature spaces and damage scenarios. A more extensive experimental analysis is recommended by employing lab specimens, on which controlled damage scenarios can be induced, and calibrated numerical models to generate simulated damaged data to validate the classifiers. Simulating damage scenarios to produce valuable information for the supervised learning could be a significant tool for high order classification and damage identification, but comparing the reliability of the simulated samples with actual records from the real structure is essential to test effectiveness and robustness of a similar approach against real-world sources of uncertainties.

Finally, it is worth reminding that the present thesis was developed within a general framework whose aim is to increase the value of SHM by exploring the potential of nature-inspired and bio-inspired strategies. The damage detection method here presented shall be cast within a cost-effective holistic SHM system together with other optimised high-level components (e.g. sensors and instruments, data fusion algorithms, feature extraction procedures, model-based damage identification methods, etc.). Therefore, more research is needed to find and test significant approaches for each task and to prove their interoperability, aiming at a common goal: an optimised sensing system, avoiding over-instrumentation, which produces a dataset large enough to be properly managed and processed, in order to generate few and clear information with high industrial and societal relevance. In this regard, the author think that biomimetics has an extreme potential to produce effective solutions and breakthrough in the field of SHM.

LIST OF REFERENCES

- [1] C. R. Farrar and K. Worden, 'Machine Learning and Statistical Pattern Recognition', in *Structural Health Monitoring: A Machine Learning Perspective*, 2012, pp. 295–320.
- [2] C. R. Farrar and K. Worden, 'Introduction', in *Structural Health Monitoring: A Machine Learning Perspective*, 2012, pp. 1–16.
- [3] U. Starossek, *Progressive collapse of structures*. Thomas Telford Publishing, 2009.
- [4] P. Cawley, 'Non-destructive testing—current capabilities and future directions', *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*, vol. 215, no. 4, pp. 213–223, Oct. 2001, doi: 10.1177/146442070121500403.
- [5] M. Moore, B. M. Phares, B. Graybeal, D. Rolander, and G. Washer, 'Reliability of Visual Inspection for Highway Bridges, Volume I: Final Report', Jun. 2001. [Online]. Available: <https://rosap.ntl.bts.gov/view/dot/33883>.
- [6] F. Moreu and J. M. LaFave, 'Current Research Topics: Railroad Bridges and Structural Engineering', Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign, Technical Report 032, Oct. 2012. Accessed: Jan. 10, 2021. [Online]. Available: <http://hdl.handle.net/2142/34749>.
- [7] E. G. Straser, A. S. Kiremidjian, T. H. Meng, and L. Redlefsen, 'A modular, wireless network platform for monitoring structures', in *Proceedings – SPIE The International Society for Optical Engineering*, issue, 1998, pp. 450–456.
- [8] Z. van Leeuwen and R. Lamb, 'Flood and scour related failure incidents at railway assets', JBA Trust, Apr. 2014. Accessed: Jan. 10, 2021. [Online].
- [9] J.-S. Tan, K. Elbaz, Z.-F. Wang, J. S. Shen, and J. Chen, 'Lessons Learnt from Bridge Collapse: A View of Sustainable Management', *Sustainability*, vol. 12, no. 3, Art. no. 3, Jan. 2020, doi: 10.3390/su12031205.
- [10] D. Pearson-Kirk, 'The benefits of bridge condition monitoring', *Proceedings of the Institution of Civil Engineers - Bridge Engineering*, vol. 161, no. 3, pp. 151–158, 2008, doi: 10.1680/bren.2008.161.3.151.
- [11] S. Zhu, D. Levinson, H. X. Liu, and K. Harder, 'The traffic and behavioral effects of the I-35W Mississippi River bridge collapse', *Transportation Research Part A: Policy and Practice*, vol. 44, no. 10, pp. 771–784, 2010, doi: <https://doi.org/10.1016/j.tra.2010.07.001>.
- [12] T. E. Mulinazzi, E. J. Fitzsimmons, S. D. Schrock, and R. Roth, 'Economic impact of closing low-volume rural bridges.', Feb. 2013, [Online]. Available: <https://rosap.ntl.bts.gov/view/dot/25697>.
- [13] D. Maharjan, M. Agüero, C. Lippitt, and F. Moreu, 'Infrastructure Stakeholders' Perspective in Development and Implementation of New Structural Health Monitoring (SHM) Technologies for Maintenance and Management of Transportation Infrastructure', *MATEC Web Conf.*, vol. 271, p. 01010, 2019, doi: 10.1051/mateconf/201927101010.
- [14] D. Straub et al., 'Value of information: A roadmap to quantifying the benefit of structural health monitoring', presented at the ICOSAR - 12th International Conference on Structural Safety & Reliability, Aug. 2017, Accessed: Jan. 10, 2021. [Online]. Available: <https://hal.inria.fr/hal-01577257>.
- [15] P. Omenzetter, U. Yazgan, S. Soyoz, and M. P. Limongelli, 'Quantifying the value of SHM for emergency management of bridges at-risk from seismic damage', in *Joint COST Actions TU1402 & TU1406 and IABSE WC1 Workshop: Quantifying the Value of Structural Health Monitoring for the Reliable Bridge Management*, 2017, pp. 4–5.

-
- [16] S. Thöns, M. H. Faber, and D. V. Val, 'On the value of structural health monitoring information for the operation of wind parks', in *Proceedings of the Safety, Reliability, Risk, Resilience and Sustainability of Structures and Infrastructure, 12th International Conference on Structural Safety and Reliability*, Wien Vienna, Austria, 2017, pp. 6–10.
 - [17] K. Papakonstantinou, C. Andriotis, H. Gao, and E. Chatzi, 'Quantifying the value of structural monitoring for decision making', May 2019, doi: 10.22725/ICASP13.382.
 - [18] S. Thöns, W. J. Klerk, and J. Köhler, 'Case studies for quantifying the value of structural health monitoring information: lessons learnt', 2019.
 - [19] C. J. Hellier, *Handbook of Nondestructive Evaluation*, Second Edition, 2nd edition. New York: McGraw-Hill Education, 2013.
 - [20] V. M. Malhotra and N. J. Carino, *Handbook on nondestructive testing of concrete*. CRC press, 2003.
 - [21] H. Czichos, *Handbook of technical diagnostics: fundamentals and application to structures and systems*. Springer Science & Business Media, 2013.
 - [22] P. J. Shull, *Nondestructive evaluation: theory, techniques, and applications*. CRC press, 2002.
 - [23] J.-P. Balayssac and V. Garnier, *Non-destructive testing and evaluation of civil engineering structures*. Elsevier, 2017.
 - [24] K. Worden, C. R. Farrar, G. Manson, and G. Park, 'The fundamental axioms of structural health monitoring', *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2082, pp. 1639–1664, Jun. 2007, doi: 10.1098/rspa.2007.1834.
 - [25] C. R. Farrar and K. Worden, 'Fundamental Axioms of Structural Health Monitoring', in *Structural Health Monitoring: A Machine Learning Perspective*, 2012, pp. 439–460.
 - [26] C. R. Farrar, T. A. Duffey, S. W. Doebling, and D. A. Nix, 'A statistical pattern recognition paradigm for vibration-based structural health monitoring', *Structural Health Monitoring*, vol. 2000, pp. 764–773, 1999.
 - [27] D. L. Hall and S. A. H. McMullen, *Mathematical techniques in multisensor data fusion*. Artech House, 2004.
 - [28] D. Bernal, '3 - Analytical techniques for damage detection and localization for assessing and monitoring civil infrastructures', in *Woodhead Publishing Series in Electronic and Optical Materials*, vol. 56, M. L. Wang, J. P. Lynch, and H. B. T.-S. T. for C. I. Sohn, Eds. Woodhead Publishing, 2014, pp. 67–92.
 - [29] C. R. Farrar and K. Worden, 'An introduction to structural health monitoring', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 303–315, 2007, doi: 10.1098/rsta.2006.1928.
 - [30] A. Rytter, 'Vibrational based inspection of civil engineering structures'. Dept. of Building Technology and Structural Engineering, Aalborg University, 1993.
 - [31] D. Zonta, '2 - Sensor data analysis, reduction and fusion for assessing and monitoring civil infrastructures', in *Woodhead Publishing Series in Electronic and Optical Materials*, vol. 56, M. L. Wang, J. P. Lynch, and H. B. T.-S. T. for C. I. Sohn, Eds. Woodhead Publishing, 2014, pp. 33–66.
 - [32] C. R. Farrar and K. Worden, 'Damage-Sensitive Features', in *Structural Health Monitoring: A Machine Learning Perspective*, 2012, pp. 161–243.
 - [33] C. R. Farrar and K. Worden, 'Data Normalisation', in *Structural Health Monitoring: A Machine Learning Perspective*, 2012, pp. 403–438.
 - [34] C. R. Farrar and K. Worden, 'Supervised Learning – Classification and Regression', in *Structural Health Monitoring: A Machine Learning Perspective*, 2012, pp. 361–401.
 - [35] W. Ostachowicz, R. Soman, and P. Malinowski, 'Optimization of sensor placement for structural health monitoring: a review', *Structural Health Monitoring*, p. 1475921719825601, Jan. 2019, doi: 10.1177/1475921719825601.
-

- [36] T. J. Matarazzo, S. G. Shahidi, M. Chang, and S. N. Pakzad, 'Are Today's SHM Procedures Suitable for Tomorrow's BIGDATA? BT - Structural Health Monitoring and Damage Detection, Volume 7', 2015, pp. 59–65.
- [37] C. Cremona, 'Big data and structural health monitoring', in *Proceedings of the 19th IABSE Congress: Challenges in Design and Construction of an Innovative and Sustainable Built Environment*, Stockholm, Sweden, 2016, pp. 21–23.
- [38] Y. Q. Ni, Y. Xia, W. Y. Liao, and J. M. Ko, 'Technology innovation in developing the structural health monitoring system for Guangzhou New TV Tower', *Structural Control and Health Monitoring*, vol. 16, no. 1, pp. 73–98, Feb. 2009, doi: 10.1002/stc.303.
- [39] K.-Y. Wong and Y.-Q. Ni, 'Modular Architecture of SHM System for Cable-Supported Bridges', *Encyclopedia of Structural Health Monitoring*. Jan. 26, 2009, doi: doi:10.1002/9780470061626.shm162.
- [40] H. Gil, J. Park, J. Cho, and G. Jung, 'Renovation of Structural Health Monitoring System for Seohae Bridge, Seoul, South Korea', *Transportation Research Record*, vol. 2201, no. 1, pp. 131–138, Jan. 2010, doi: 10.3141/2201-15.
- [41] O. Riches, C. Hill, and P. Baralos, 'Queensferry Crossing, UK: durability, maintenance, inspection and monitoring', *Proceedings of the Institution of Civil Engineers - Bridge Engineering*, vol. 172, no. 2, pp. 175–188, 2019, doi: 10.1680/jbren.18.00020.
- [42] R. A. B. Jr., K. A. Grimmelmsan, and A. E. Aktan, 'Instrumented monitoring of the Commodore Barry Bridge', in *Proc.SPIE*, Jun. 2000, vol. 3995.
- [43] P. Papanikolas, A. Stathopoulos-Vlamiis, and A. Panagis, 'The structural health monitoring system of the Rion Antirion Bridge "Charilaos Trikoupis"'.
- [44] C. Cremona and J. Santos, 'Structural Health Monitoring as a Big-Data Problem', *Structural Engineering International*, vol. 28, no. 3, pp. 243–254, Jul. 2018, doi: 10.1080/10168664.2018.1461536.
- [45] C. R. Farrar, G. Park, D. W. Allen, and M. D. Todd, 'Sensor network paradigms for structural health monitoring', *Structural Control and Health Monitoring*, vol. 13, no. 1, pp. 210–225, Jan. 2006, doi: 10.1002/stc.125.
- [46] G. W. Housner et al., 'Structural Control: Past, Present, and Future', *Journal of Engineering Mechanics*, vol. 123, no. 9, pp. 897–971, Sep. 1997, doi: 10.1061/(ASCE)0733-9399(1997)123:9(897).
- [47] D. Lowe, 'Feature extraction, data visualisation, classification and fusion for damage assessment', 2000.
- [48] M. G. Masciotta, A. Barontini, L. F. Ramos, P. Amado-Mendes, and P. B. Lourenço, 'An overview on structural health monitoring: From the current state-of-the-art to new bio-inspired sensing paradigms', *International Journal of Bio-Inspired Computation*, vol. 14, no. 1, 2019, doi: 10.1504/IJBIC.2019.101179.
- [49] M.-G. Masciotta, A. Barontini, L. F. Ramos, P. Amado-Mendes, and P. B. Lourenço, 'A Bio-inspired Framework for Highly Efficient Structural Health Monitoring and Vibration Analysis', 2017.
- [50] A. Barontini, M.-G. Masciotta, L. F. Ramos, P. Amado-Mendes, and P. B. Lourenço, 'An overview on nature-inspired optimization algorithms for Structural Health Monitoring of historical buildings', *Procedia Engineering*, vol. 199, pp. 3320–3325, 2017, doi: https://doi.org/10.1016/j.proeng.2017.09.439.
- [51] G. Capellari, E. Chatzi, and S. Mariani, 'An optimal sensor placement method for SHM based on Bayesian experimental design and Polynomial Chaos Expansion', in *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*, 2016, vol. 3, pp. 6272–6282, doi: 10.7712/100016.2257.6762.

-
- [52] M. L. Wang, J. P. Lynch, and H. Sohn, '1 - Introduction to sensing for structural performance assessment and health monitoring', in Woodhead Publishing Series in Electronic and Optical Materials, vol. 55, M. L. Wang, J. P. Lynch, and H. B. T.-S. T. for C. I. Sohn, Eds. Woodhead Publishing, 2014, pp. 1–22.
- [53] 'Unsupervised Learning – Novelty Detection', Structural Health Monitoring. pp. 321–360, Nov. 22, 2012, doi: doi:10.1002/9781118443118.ch10.
- [54] 'Operational Evaluation', Structural Health Monitoring. pp. 45–52, Nov. 22, 2012, doi: doi:10.1002/9781118443118.ch3.
- [55] N. S. Gulgec, G. S. Shahidi, T. J. Matarazzo, and S. N. Pakzad, 'Current Challenges with BIGDATA Analytics in Structural Health Monitoring BT - Structural Health Monitoring & Damage Detection, Volume 7', 2017, pp. 79–84.
- [56] D. Laney, '3D data management: Controlling data volume, velocity and variety', META group research note, vol. 6, no. 70, p. 1, 2001.
- [57] M. Hilbert, 'Big Data for Development: A Review of Promises and Challenges', Development Policy Review, vol. 34, no. 1, pp. 135–174, Jan. 2016, doi: 10.1111/dpr.12142.
- [58] Y. Bao, Z. Tang, H. Li, and Y. Zhang, 'Computer vision and deep learning-based data anomaly detection method for structural health monitoring', Structural Health Monitoring, vol. 18, no. 2, pp. 401–421, Feb. 2018, doi: 10.1177/1475921718757405.
- [59] M. T. J. and P. S. N., 'Truncated Physical Model for Dynamic Sensor Networks with Applications in High-Resolution Mobile Sensing and BIGDATA', Journal of Engineering Mechanics, vol. 142, no. 5, p. 4016019, May 2016, doi: 10.1061/(ASCE)EM.1943-7889.0001022.
- [60] M. T. J. and P. S. N., 'STRIDE for Structural Identification Using Expectation Maximization: Iterative Output-Only Method for Modal Identification', Journal of Engineering Mechanics, vol. 142, no. 4, p. 4015109, Apr. 2016, doi: 10.1061/(ASCE)EM.1943-7889.0000951.
- [61] Z. Feng, Q. Wang, and K. Shida, 'A review of self-validating sensor technology', Sensor Review, vol. 27, no. 1, pp. 48–56, 2007.
- [62] Z. Shen and Q. Wang, 'Failure Detection, Isolation, and Recovery of Multifunctional Self-Validating Sensor', IEEE Transactions on Instrumentation and Measurement, vol. 61, no. 12, pp. 3351–3362, 2012, doi: 10.1109/TIM.2012.2205509.
- [63] B. F. Spencer Jr, J.-W. Park, K. A. Mechitov, H. Jo, and G. Agha, 'Next generation wireless smart sensors toward sustainable civil infrastructure', Procedia Engineering, vol. 171, pp. 5–13, 2017.
- [64] H. He, W. Yan, and A. Zhang, 'Structural Damage Information Fusion Based on Soft Computing', International Journal of Distributed Sensor Networks, vol. 8, no. 9, p. 798714, Sep. 2012, doi: 10.1155/2012/798714.
- [65] Y. Ding and A. Li, 'Temperature-induced variations of measured modal frequencies of steel box girder for a long-span suspension bridge', International Journal of Steel Structures, vol. 11, no. 2, pp. 145–155, 2011, doi: 10.1007/s13296-011-2004-4.
- [66] I. Laory, T. N. Trinh, I. F. C. Smith, and J. M. W. Brownjohn, 'Methodologies for predicting natural frequency variation of a suspension bridge', Engineering Structures, vol. 80, pp. 211–221, 2014, doi: https://doi.org/10.1016/j.engstruct.2014.09.001.
- [67] N. Y. Q., Z. H. F., and K. J. M., 'Generalization Capability of Neural Network Models for Temperature-Frequency Correlation Using Monitoring Data', Journal of Structural Engineering, vol. 135, no. 10, pp. 1290–1300, Oct. 2009, doi: 10.1061/(ASCE)ST.1943-541X.0000050.
- [68] N. Martins, E. Caetano, S. Diord, F. Magalhães, and Á. Cunha, 'Dynamic monitoring of a stadium suspension roof: Wind and temperature influence on modal parameters and structural response', Engineering Structures, vol. 59, pp. 80–94, 2014, doi: https://doi.org/10.1016/j.engstruct.2013.10.021.
-

- [69] L. Faravelli, F. Ubertini, and C. Fuggini, 'System identification of a super high-rise building via a stochastic subspace approach', vol. 7, no. 2, pp. 133–152, Feb. 2011, doi: 10.12989/SSS.2011.7.2.133.
- [70] W. H. Chen et al., 'Theoretical and experimental modal analysis of the Guangzhou New TV Tower', *Engineering Structures*, vol. 33, no. 12, pp. 3628–3646, 2011, doi: <https://doi.org/10.1016/j.engstruct.2011.07.028>.
- [71] M.-G. Masciotta, J. C. A. Roque, L. F. Ramos, and P. B. Lourenço, 'A multidisciplinary approach to assess the health state of heritage structures: The case study of the Church of Monastery of Jerónimos in Lisbon', *Construction and Building Materials*, vol. 116, pp. 169–187, 2016, doi: <http://dx.doi.org/10.1016/j.conbuildmat.2016.04.146>.
- [72] L. F. Ramos, L. Marques, P. B. Lourenço, G. De Roeck, A. Campos-Costa, and J. Roque, 'Monitoring historical masonry structures with operational modal analysis: Two case studies', *Mechanical Systems and Signal Processing*, vol. 24, no. 5, pp. 1291–1305, 2010, doi: <https://doi.org/10.1016/j.ymssp.2010.01.011>.
- [73] C. Gentile and A. Saisi, 'Ambient vibration testing of historic masonry towers for structural identification and damage assessment', *Construction and Building Materials*, vol. 21, no. 6, pp. 1311–1321, 2007, doi: <https://doi.org/10.1016/j.conbuildmat.2006.01.007>.
- [74] C. Gentile, A. Saisi, and A. Cabboi, 'Dynamic monitoring of a masonry tower', 2012.
- [75] C. Gentile, M. Guidobaldi, and A. Saisi, 'One-year dynamic monitoring of a historic tower: damage detection under changing environment', *Meccanica*, vol. 51, no. 11, pp. 2873–2889, 2016, doi: 10.1007/s11012-016-0482-3.
- [76] A. Saisi, C. Gentile, and M. Guidobaldi, 'Post-earthquake continuous dynamic monitoring of the Gabbia Tower in Mantua, Italy', *Construction and Building Materials*, vol. 81, pp. 101–112, 2015, doi: <https://doi.org/10.1016/j.conbuildmat.2015.02.010>.
- [77] C. Tsogka, E. Daskalakis, G. Comanducci, and F. Ubertini, 'The Stretching Method for Vibration-Based Structural Health Monitoring of Civil Structures', *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 4, pp. 288–303, Apr. 2017, doi: 10.1111/mice.12255.
- [78] F. Ubertini, G. Comanducci, and N. Cavalagli, 'Vibration-based structural health monitoring of a historic bell-tower using output-only measurements and multivariate statistical analysis', *Structural Health Monitoring*, vol. 15, no. 4, pp. 438–457, May 2016, doi: 10.1177/1475921716643948.
- [79] F. Ubertini, G. Comanducci, N. Cavalagli, A. Laura Pisello, A. Luigi Materazzi, and F. Cotana, 'Environmental effects on natural frequencies of the San Pietro bell tower in Perugia, Italy, and their removal for structural performance assessment', *Mechanical Systems and Signal Processing*, vol. 82, pp. 307–322, 2017, doi: <https://doi.org/10.1016/j.ymssp.2016.05.025>.
- [80] R. Mario Azzara, G. De Roeck, M. Girardi, C. Padovani, D. Pellegrini, and E. Reynders, 'The influence of environmental parameters on the dynamic behaviour of the San Frediano bell tower in Lucca', *Engineering Structures*, vol. 156, pp. 175–187, 2018, doi: <https://doi.org/10.1016/j.engstruct.2017.10.045>.
- [81] B. Peeters and G. De Roeck, 'One-year monitoring of the Z24-Bridge: environmental effects versus damage events', *Earthquake Engineering & Structural Dynamics*, vol. 30, no. 2, pp. 149–171, Feb. 2001, doi: 10.1002/1096-9845(200102)30:2<149::AID-EQE1>3.0.CO;2-Z.
- [82] A. Cury, C. Cremona, and J. Dumoulin, 'Long-term monitoring of a PSC box girder bridge: Operational modal analysis, data normalization and structural modification assessment', *Mechanical Systems and Signal Processing*, vol. 33, pp. 13–37, 2012, doi: <https://doi.org/10.1016/j.ymssp.2012.07.005>.
- [83] G.-D. Zhou and T.-H. Yi, 'A summary review of correlations between temperatures and vibration properties of long-span bridges', *Mathematical Problems in Engineering*, vol. 2014, 2014.

-
- [84] H. Sohn, K. Worden, and C. R. Farrar, 'Statistical Damage Classification Under Changing Environmental and Operational Conditions', *Journal of Intelligent Material Systems and Structures*, vol. 13, no. 9, pp. 561–574, Sep. 2002, doi: 10.1106/104538902030904.
 - [85] A.-M. Yan, G. Kerschen, P. De Boe, and J.-C. Golinval, 'Structural damage diagnosis under varying environmental conditions—Part I: A linear analysis', *Mechanical Systems and Signal Processing*, vol. 19, no. 4, pp. 847–864, 2005, doi: <https://doi.org/10.1016/j.ymssp.2004.12.002>.
 - [86] A.-M. Yan, G. Kerschen, P. De Boe, and J.-C. Golinval, 'Structural damage diagnosis under varying environmental conditions—part II: local PCA for non-linear cases', *Mechanical Systems and Signal Processing*, vol. 19, no. 4, pp. 865–880, 2005, doi: <https://doi.org/10.1016/j.ymssp.2004.12.003>.
 - [87] H. Sohn, 'Effects of environmental and operational variability on structural health monitoring', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, p. 539 LP – 560, Feb. 2007.
 - [88] V. M. W., B. J. L., and A. S. K., 'Bayesian Probabilistic Approach to Structural Health Monitoring', *Journal of Engineering Mechanics*, vol. 126, no. 7, pp. 738–745, Jul. 2000, doi: 10.1061/(ASCE)0733-9399(2000)126:7(738).
 - [89] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.
 - [90] K. Worden and G. Manson, 'The application of machine learning to structural health monitoring', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, p. 515 LP – 537, Feb. 2007.
 - [91] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
 - [92] C. Blum and A. Roli, 'Metaheuristics in combinatorial optimization: Overview and conceptual comparison', *ACM COMPUTING SURVEYS*, pp. 268–308, 2003.
 - [93] S. I. F. and S. Sandro, 'Improving Knowledge of Structural System Behavior through Multiple Models', *Journal of Structural Engineering*, vol. 134, no. 4, pp. 553–561, Apr. 2008, doi: 10.1061/(ASCE)0733-9445(2008)134:4(553).
 - [94] C. Robert, 'Machine Learning, a Probabilistic Perspective', *CHANCE*, vol. 27, no. 2, pp. 62–63, Apr. 2014, doi: 10.1080/09332480.2014.914768.
 - [95] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. The MIT Press, 2010.
 - [96] K. Worden, H. Sohn, and C. R. Farrar, 'NOVELTY DETECTION IN A CHANGING ENVIRONMENT: REGRESSION AND INTERPOLATION APPROACHES', *Journal of Sound and Vibration*, vol. 258, no. 4, pp. 741–761, 2002, doi: <http://dx.doi.org/10.1006/jsvi.2002.5148>.
 - [97] A. Bellino, A. Fasana, L. Garibaldi, and S. Marchesiello, 'PCA-based detection of damage in time-varying systems', *Mechanical Systems and Signal Processing*, vol. 24, no. 7, pp. 2250–2260, 2010, doi: <http://dx.doi.org/10.1016/j.ymssp.2010.04.009>.
 - [98] Y. Kim, J. W. Chong, K. H. Chon, and J. Kim, 'Wavelet-based AR–SVM for health monitoring of smart structures', *Smart Materials and Structures*, vol. 22, no. 1, p. 15003, 2012, doi: 10.1088/0964-1726/22/1/015003.
 - [99] O. R. de Lautour and P. Omenzetter, 'Damage classification and estimation in experimental structures using time series analysis and pattern recognition', *Mechanical Systems and Signal Processing*, vol. 24, no. 5, pp. 1556–1569, 2010, doi: <https://doi.org/10.1016/j.ymssp.2009.12.008>.
 - [100] F. Gonzalez, D. Dasgupta, and R. Kozma, 'Combining negative selection and classification techniques for anomaly detection', *Evolutionary Computation*, 2002. CEC '02. Proceedings of the 2002 Congress on, vol. 1, pp. 705–710, 2002, doi: 10.1109/CEC.2002.1007012.
 - [101] K. WORDEN, G. MANSON, and N. R. J. FIELLER, 'DAMAGE DETECTION USING OUTLIER ANALYSIS', *Journal of Sound and Vibration*, vol. 229, no. 3, pp. 647–667, 2000, doi: <https://doi.org/10.1006/jsvi.1999.2514>.
-

- [102] J. P. de Oliveira Dias Prudente dos Santos, C. Crémona, A. P. C. da Silveira, and L. C. de Oliveira Martins, 'Real-time damage detection based on pattern recognition', *Structural Concrete*, vol. 17, no. 3, pp. 338–354, Sep. 2016, doi: 10.1002/suco.201500092.
- [103] J. P. Santos, C. Crémona, L. Calado, P. Silveira, and A. D. Orcesi, 'On-line unsupervised detection of early damage', *Structural Control and Health Monitoring*, vol. 23, no. 7, pp. 1047–1069, Jul. 2016, doi: 10.1002/stc.1825.
- [104] J. P. Santos, A. D. Orcesi, C. Crémona, and P. Silveira, 'Baseline-free real-time assessment of structural changes', *Structure and Infrastructure Engineering*, vol. 11, no. 2, pp. 145–161, Feb. 2015, doi: 10.1080/15732479.2013.858169.
- [105] K. Krishnan Nair and A. S. Kiremidjian, 'Time Series Based Structural Damage Detection Algorithm Using Gaussian Mixtures Modeling', *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 3, pp. 285–293, Aug. 2006, doi: 10.1115/1.2718241.
- [106] K. N. Kesavan and A. S. Kiremidjian, 'A wavelet-based damage diagnosis algorithm using principal component analysis', *Structural Control and Health Monitoring*, vol. 19, no. 8, pp. 672–685, Dec. 2012, doi: 10.1002/stc.462.
- [107] J. F. V Vincent, O. A. Bogatyreva, N. R. Bogatyrev, A. Bowyer, and A.-K. Pahl, 'Biomimetics: its practice and theory', *Journal of The Royal Society Interface*, vol. 3, no. 9, pp. 471–482, Aug. 2006, doi: 10.1098/rsif.2006.0127.
- [108] B. Bhushan, 'Biomimetics: lessons from nature—an overview', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1893, pp. 1445–1486, Apr. 2009, doi: 10.1098/rsta.2009.0011.
- [109] K. J. Loh, D. Ryu, and B. M. Lee, 'Bio-inspired Sensors for Structural Health Monitoring BT - Biotechnologies and Biomimetics for Civil Engineering', F. Pacheco Torgal, J. A. Labrincha, M. V. Diamanti, C.-P. Yu, and H. K. Lee, Eds. Cham: Springer International Publishing, 2015, pp. 255–274.
- [110] U. Tata, S. Deshmukh, J. C. Chiao, R. Carter, and H. Huang, 'Bio-inspired sensor skins for structural health monitoring', *Smart Materials and Structures*, vol. 18, no. 10, p. 104026, 2009.
- [111] C. A. Peckens and J. P. Lynch, 'Utilizing the cochlea as a bio-inspired compressive sensing technique', *Smart Materials and Structures*, vol. 22, no. 10, p. 105027, 2013, doi: 10.1088/0964-1726/22/10/105027.
- [112] N. Salowitz et al., 'Recent advancements and vision toward stretchable bio-inspired networks for intelligent structures', *Structural Health Monitoring*, vol. 13, no. 6, pp. 609–620, Nov. 2014, doi: 10.1177/0725513614554076.
- [113] L. N. de Castro, 'Fundamentals of natural computing: an overview', *Physics of Life Reviews*, vol. 4, no. 1, pp. 1–36, 2007, doi: <http://dx.doi.org/10.1016/j.plrev.2006.10.002>.
- [114] G. Rozenberg, 'Computer Science, Informatics, and Natural Computing—Personal Reflections BT - New Computational Paradigms: Changing Conceptions of What is Computable', S. B. Cooper, B. Löwe, and A. Sorbi, Eds. New York, NY: Springer New York, 2008, pp. 373–379.
- [115] L. O. Chua, *CNN: A paradigm for complexity*, vol. 31. World Scientific, 1998.
- [116] B. J. MacLennan, 'Natural computation and non-Turing models of computation', *Theoretical Computer Science*, vol. 317, no. 1, pp. 115–145, 2004, doi: <https://doi.org/10.1016/j.tcs.2003.12.008>.
- [117] G. Păun, 'From cells to (silicon) computers, and back', in *New computational paradigms*, Springer, 2008, pp. 343–371.
- [118] J. N. Hooker, 'Testing heuristics: We have it all wrong', *Journal of Heuristics*, vol. 1, no. 1, pp. 33–42, 1995, doi: 10.1007/BF02430364.
- [119] C. A. Janeway, P. Travers, M. Walport, and M. J. Shlomchik, 'Immunobiology: the immune system in health and disease', 2005.

-
- [120] S. M. Garrett, 'How do we evaluate artificial immune systems?', *Evolutionary computation*, vol. 13, no. 2, pp. 145–177, 2005.
- [121] P. J. C. Branco, J. A. Dente, and R. V. Mendes, 'Using immunology principles for fault detection', *IEEE Transactions on Industrial Electronics*, vol. 50, no. 2, pp. 362–373, Apr. 2003, doi: 10.1109/TIE.2003.809418.
- [122] M. Ebner, H.-G. Breunig, and J. Albert, 'On the use of negative selection in an artificial immune system', in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 2002, pp. 957–964.
- [123] H. Bersini, 'Self-assertion versus self-recognition: A tribute to Francisco Varela', 2002.
- [124] S. P. Coy, B. L. Golden, G. C. Runger, and E. A. Wasil, 'Using Experimental Design to Find Effective Parameter Settings for Heuristics', *Journal of Heuristics*, vol. 7, no. 1, pp. 77–97, 2001, doi: 10.1023/A:1026569813391.
- [125] T.-H. Yi and H.-N. Li, 'Methodology Developments in Sensor Placement for Health Monitoring of Civil Infrastructures', *International Journal of Distributed Sensor Networks*, vol. 8, no. 8, p. 612726, Jan. 2012, doi: 10.1155/2012/612726.
- [126] J. L. and L. H. and B. M. and H. L. and W. Peng, 'Optimal sensor placement for large structures using the nearest neighbour index and a hybrid swarm intelligence algorithm', *Smart Materials and Structures*, vol. 22, no. 9, p. 95015, 2013.
- [127] T. You, H. Jin, and P. Li, 'Optimal Placement of Wireless Sensor Nodes for Bridge Dynamic Monitoring Based on Improved Particle Swarm Algorithm', *International Journal of Distributed Sensor Networks*, vol. 9, no. 12, p. 390936, Jan. 2013, doi: 10.1155/2013/390936.
- [128] G.-D. Zhou, T.-H. Yi, and H.-N. Li, 'Sensor Placement Optimization in Structural Health Monitoring Using Cluster-In-Cluster Firefly Algorithm', *Advances in Structural Engineering*, vol. 17, no. 8, pp. 1103–1115, Aug. 2014, doi: 10.1260/1369-4332.17.8.1103.
- [129] H. S. and O. Büyüköztürk, 'Optimal sensor placement in structural health monitoring using discrete optimization', *Smart Materials and Structures*, vol. 24, no. 12, p. 125034, 2015.
- [130] H. Jin, J. Xia, and Y. Wang, 'Optimal sensor placement for space modal identification of crane structures based on an improved harmony search algorithm', *Journal of Zhejiang University-SCIENCE A*, vol. 16, no. 6, pp. 464–477, 2015, doi: 10.1631/jzus.A1400363.
- [131] T.-H. Yi, H.-N. Li, and C.-W. Wang, 'Multiaxial sensor placement optimization in structural health monitoring using distributed wolf algorithm', *Structural Control and Health Monitoring*, vol. 23, no. 4, pp. 719–734, 2016, doi: 10.1002/stc.1806.
- [132] L. He, J. Lian, B. Ma, and H. Wang, 'Optimal multiaxial sensor placement for modal identification of large structures', *Structural Control and Health Monitoring*, vol. 21, no. 1, pp. 61–79, 2014, doi: 10.1002/stc.1550.
- [133] J. J. and S. F. and W. Liu, 'A triaxial accelerometer monkey algorithm for optimal sensor placement in structural health monitoring', *Measurement Science and Technology*, vol. 26, no. 6, p. 65104, 2015.
- [134] X.-H. Zhang, Y.-L. Xu, S. Zhu, and S. Zhan, 'Dual-type sensor placement for multi-scale response reconstruction', *Mechatronics*, vol. 24, no. 4, pp. 376–384, 2014, doi: <http://dx.doi.org/10.1016/j.mechatronics.2013.05.007>.
- [135] W. M. Ostachowicz, M. Krawczuk, and M. P. Cartmell, 'Genetic algorithms in health monitoring of structures', *Lecture Notes-AMAS*, 2001.
- [136] S. Gerist and M. R. Maheri, 'Multi-stage approach for structural damage detection problem using basis pursuit and particle swarm optimization', *Journal of Sound and Vibration*, vol. 384, pp. 210–226, 2016, doi: <http://dx.doi.org/10.1016/j.jsv.2016.08.024>.
-

- [137] B. Nanda, D. Maity, and D. K. Maiti, 'Damage assessment from curvature mode shape using unified particle swarm optimization', *Structural Engineering and Mechanics*, vol. 52, no. 2, pp. 307–322, 2014, doi: 10.12989/sem.2014.52.2.307.
- [138] L. F. Fadel Miguel, R. Holdorf Lopez, and L. F. Fadel Miguel, 'A hybrid approach for damage detection of structures under operational conditions', *Journal of Sound and Vibration*, vol. 332, no. 18, pp. 4241–4260, 2013, doi: <http://dx.doi.org/10.1016/j.jsv.2013.03.017>.
- [139] M. Masoumi, E. Jamshidi, and M. Bamdad, 'Application of generalized flexibility matrix in damage identification using Imperialist Competitive Algorithm', *KSCE Journal of Civil Engineering*, vol. 19, no. 4, pp. 994–1001, 2015, doi: 10.1007/s12205-015-0224-4.
- [140] M. M. Saada, M. H. Arafa, and A. O. Nassef, 'Finite element model updating approach to damage identification in beams using particle swarm optimization', *Engineering Optimization*, vol. 45, no. 6, pp. 677–696, 2013, doi: 10.1080/0305215X.2012.704026.
- [141] R. Perera, S.-E. Fang, and A. Ruiz, 'Application of particle swarm optimization and genetic algorithms to multiobjective damage identification inverse problems with modelling errors', *Meccanica*, vol. 45, no. 5, pp. 723–734, 2010, doi: 10.1007/s11012-009-9264-5.
- [142] S. Z. Mehrian, S. A. R. Amrei, M. Maniat, and S. M. Nowruzpour, 'Structural health monitoring using optimising algorithms based on flexibility matrix approach and combination of natural frequencies and mode shapes', *International Journal of Structural Engineering*, vol. 7, no. 4, pp. 398–411, 2016.
- [143] L. Yu and X. Chen, 'Bridge damage identification by combining modal flexibility and PSO methods', in *2010 Prognostics and System Health Management Conference*, 2010, pp. 1–6, doi: 10.1109/PHM.2010.5413488.
- [144] R. Perera and A. Ruiz, 'A multistage FE updating procedure for damage identification in large-scale structures based on multiobjective evolutionary optimization', *Mechanical Systems and Signal Processing*, vol. 22, no. 4, pp. 970–991, 2008, doi: <http://dx.doi.org/10.1016/j.ymssp.2007.10.004>.
- [145] S. M. Seyedpoor, 'A two stage method for structural damage detection using a modal strain energy based index and particle swarm optimization', *International Journal of Non-Linear Mechanics*, vol. 47, no. 1, pp. 1–8, 2012, doi: <https://doi.org/10.1016/j.ijnonlinmec.2011.07.011>.
- [146] Z. Chen and L. Yu, 'A novel two-step intelligent algorithm for damage detection of beam-like structures', in *2015 11th International Conference on Natural Computation (ICNC)*, 2015, pp. 633–638, doi: 10.1109/ICNC.2015.7378063.
- [147] S.-L. Ma, S.-F. Jiang, and L.-Q. Weng, 'Two-Stage Damage Identification Based on Modal Strain Energy and Revised Particle Swarm Optimization', *International Journal of Structural Stability and Dynamics*, vol. 14, no. 05, p. 1440005, 2014, doi: 10.1142/S0219455414400057.
- [148] A. Kaveh, S. M. Javadi, and M. Maniat, 'Damage assessment via modal data with a mixed particle swarm strategy, ray optimizer, and harmony search', *Asian Journal of Civil Engineering*, vol. 15, no. 1, pp. 95–106, 2014.
- [149] R. A. Carlin and E. Garcia, 'Parameter optimization of a genetic algorithm for structural damage detection', in *Proceedings of the 14th International Modal Analysis Conference*, 1996, vol. 2768, p. 1292.
- [150] F. T. K. Au, Y. S. Cheng, L. G. Tham, and Z. Z. Bai, 'Structural damage detection based on a micro-genetic algorithm using incomplete and noisy modal test data', *Journal of Sound and Vibration*, vol. 259, no. 5, pp. 1081–1094, 2003, doi: <http://dx.doi.org/10.1006/jsvi.2002.5116>.
- [151] J. E. Laier and J. D. Villalba, 'Influence of incomplete modal data on the performance of a vibration-based damage detection methodology', *Civil-Comp Proceedings*, vol. 99, 2012.

-
- [152] C. C. H. Borges, H. J. C. Barbosa, and A. C. C. Lemonge, 'A structural damage identification method based on genetic algorithm and vibrational data', *International Journal for Numerical Methods in Engineering*, vol. 69, no. 13, pp. 2663–2686, 2007, doi: 10.1002/nme.1857.
- [153] K. C. Yap and D. C. Zimmerman, 'The Effect of Coding in Genetic Algorithm Based Structural Damage Detection', in *Proceedings of the 16th International Modal Analysis Conference*, 1998, vol. 3243, p. 165.
- [154] M. I. Friswell, J. E. T. Penny, and S. D. Garvey, 'A combined genetic and eigensensitivity algorithm for the location of damage in structures', *Computers & Structures*, vol. 69, no. 5, pp. 547–556, 1998, doi: [http://dx.doi.org/10.1016/S0045-7949\(98\)00125-4](http://dx.doi.org/10.1016/S0045-7949(98)00125-4).
- [155] S. Jung, S.-Y. Ok, and J. Song, 'Robust structural damage identification based on multi-objective optimization', *International Journal for Numerical Methods in Engineering*, vol. 81, no. 6, pp. 786–804, 2010, doi: 10.1002/nme.2726.
- [156] A. Barontini, M. G. Masciotta, P. Amado-Mendes, L. F. Ramos, and P. B. Lourenço, 'Particle Swarm Optimization for damage identification in beam-like structures', in *Life Cycle Analysis and Assessment in Civil Engineering: Towards an Integrated Vision: Proceedings of the Sixth International Symposium on Life-Cycle Civil Engineering (IALCCE 2018)*, 28-31 October 2018, Ghent, Belgium, 2018.
- [157] S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. Shevitz, 'Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review', United States, 1996. doi: 10.2172/249299.
- [158] H. Sohn, C. R. Farrar, F. M. Hemez, and J. J. Czarnecki, 'A Review of Structural Health Review of Structural Health Monitoring Literature 1996-2001.', 2002.
- [159] W. S. McCulloch and W. Pitts, 'A logical calculus of the ideas immanent in nervous activity', *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943, doi: 10.1007/BF02478259.
- [160] C. M. Bishop, 'Neural networks and their applications', *Review of Scientific Instruments*, vol. 65, no. 6, pp. 1803–1832, Jun. 1994, doi: 10.1063/1.1144830.
- [161] G. Yen and M. Kwak, 'Neural Network Approach for the Damage Detection of Structures', in *34th Structures, Structural Dynamics and Materials Conference*, 1993, p. 1485.
- [162] P. Z. Szewczyk and P. Hajel, 'Damage Detection in Structures Based on Feature-Sensitive Neural Networks', *Journal of Computing in Civil Engineering*, vol. 8, no. 2, pp. 163–178, Apr. 1994, doi: 10.1061/(ASCE)0887-3801(1994)8:2(163).
- [163] S.-F. Jiang, C. Fu, and C. Zhang, 'A hybrid data-fusion system using modal data and probabilistic neural network for damage detection', *Advances in Engineering Software*, vol. 42, no. 6, pp. 368–374, 2011, doi: <http://dx.doi.org/10.1016/j.advengsoft.2011.03.002>.
- [164] S. E. Klenke and T. L. Paez, 'Damage identification with probabilistic neural networks', 1995.
- [165] A. Rytter and P. Kirkegaard, 'Vibration based inspection using neural networks', *Structural Damage Assessment Using Advanced Signal Processing Procedures*, *Proceedings of DAMAS '97*, University of Sheffield, UK, pp. 97–108, 1997.
- [166] J. B. Butcher, C. R. Day, J. C. Austin, P. W. Haycock, D. Verstraeten, and B. Schrauwen, 'Defect Detection in Reinforced Concrete Using Random Neural Architectures', *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 3, pp. 191–207, 2014, doi: 10.1111/mice.12039.
- [167] M. T. Hagan, H. B. Demuth, M. Beale, and O. De Jesús, *Neural network design*. Martin Hagan, 2014.
- [168] D. F. Specht, 'Probabilistic neural networks', *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990, doi: [http://dx.doi.org/10.1016/0893-6080\(90\)90049-Q](http://dx.doi.org/10.1016/0893-6080(90)90049-Q).
- [169] R. Hecht-Nielsen, 'Applications of counterpropagation networks', *Neural Networks*, vol. 1, no. 2, pp. 131–139, 1988, doi: [http://dx.doi.org/10.1016/0893-6080\(88\)90015-9](http://dx.doi.org/10.1016/0893-6080(88)90015-9).
-

- [170] S. V Barai and P. C. Pandey, 'Time-delay neural networks in damage detection of railway bridges', *Advances in Engineering Software*, vol. 28, no. 1, pp. 1–10, 1997, doi: [http://dx.doi.org/10.1016/S0965-9978\(96\)00038-5](http://dx.doi.org/10.1016/S0965-9978(96)00038-5).
- [171] P. Li, 'Structural damage localization using probabilistic neural networks', *Mathematical and Computer Modelling*, vol. 54, no. 3, pp. 965–969, 2011, doi: <http://dx.doi.org/10.1016/j.mcm.2010.11.023>.
- [172] J. P. Amezcua-Sanchez and H. Adeli, 'Feature extraction and classification techniques for health monitoring of structures', *Scientia Iranica. Transaction A, Civil Engineering*, vol. 22, no. 6, p. 1931, 2015.
- [173] E. P. Carden and P. Fanning, 'Vibration Based Condition Monitoring: A Review', *Structural Health Monitoring*, vol. 3, no. 4, pp. 355–377, Dec. 2004, doi: 10.1177/1475921704047500.
- [174] W. Fan and P. Qiao, 'Vibration-based Damage Identification Methods: A Review and Comparative Study', *Structural Health Monitoring*, vol. 10, no. 1, pp. 83–111, Apr. 2010, doi: 10.1177/1475921710365419.
- [175] M. Y. Choi and I. B. Kwon, 'Damage detection system of a real steel truss bridge by neural networks', *bridge structures*, vol. 7, p. 13, 2000.
- [176] N. Bakhary, H. Hao, and A. J. Deeks, 'Structure Damage Detection Using Neural Network with Multi-Stage Substructuring', *Advances in Structural Engineering*, vol. 13, no. 1, pp. 95–110, Feb. 2010, doi: 10.1260/1369-4332.13.1.95.
- [177] N. Bakhary, H. Hao, and A. J. Deeks, 'Substructuring Technique for Damage Detection Using Statistical Multi-Stage Artificial Neural Network', *Advances in Structural Engineering*, vol. 13, no. 4, pp. 619–639, Aug. 2010, doi: 10.1260/1369-4332.13.4.619.
- [178] J. P. Santos, C. Cr mona, A. D. Orcesi, and P. Silveira, 'Multivariate statistical analysis for early damage detection', *Engineering Structures*, vol. 56, pp. 273–285, 2013, doi: <https://doi.org/10.1016/j.engstruct.2013.05.022>.
- [179] J. Santos, L. Calado, P. Silveira, and C. Cr mona, 'Baseline-Based Detection of Structural Changes: Drawbacks and an Alternative Approach', in *EWSHM - 7th European Workshop on Structural Health Monitoring*, 2014.
- [180] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, 'Self-Nonself Discrimination in a Computer', in *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, 1994, pp. 202–.
- [181] Z. Ji and D. Dasgupta, 'Applicability Issues of the Real-valued Negative Selection Algorithms', in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, 2006, pp. 111–118, doi: 10.1145/1143997.1144017.
- [182] D. Dasgupta, S. Yu, and F. Nino, 'Recent Advances in Artificial Immune Systems: Models and Applications', *Applied Soft Computing*, vol. 11, no. 2, pp. 1574–1587, 2011, doi: <http://dx.doi.org/10.1016/j.asoc.2010.08.024>.
- [183] Z. Ji and D. Dasgupta, 'Revisiting Negative Selection Algorithms', *Evol. Comput.*, vol. 15, no. 2, pp. 223–251, 2007, doi: 10.1162/evco.2007.15.2.223.
- [184] T. Stibor, P. Mohr, J. Timmis, and C. Eckert, 'Is Negative Selection Appropriate for Anomaly Detection?', in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, 2005, pp. 321–328, doi: 10.1145/1068009.1068061.
- [185] T. Stibor and J. Timmis, 'Comments on real-valued negative selection vs. real-valued positive selection and one-class SVM', in *2007 IEEE Congress on Evolutionary Computation*, 2007, pp. 3727–3734, doi: 10.1109/CEC.2007.4424956.
- [186] F. Esponda, S. Forrest, and P. Helman, 'A formal framework for positive and negative detection schemes', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 357–373, 2004.

-
- [187] T. Stibor, J. Timmis, and C. Eckert, 'A comparative study of real-valued negative selection to statistical anomaly detection techniques', in *International Conference on Artificial Immune Systems*, 2005, pp. 262–275.
 - [188] C. Ramdane and S. Chikhi, 'Negative selection algorithm: recent improvements and its application in intrusion detection system', *Int. J. Comput. Acad. Res.(IJCAR)*, vol. 6, no. 2, pp. 20–30, 2017.
 - [189] M. F. Mohamad Mohsin, A. R. Hamdan, and A. Abu Bakar, 'The Effect of Normalization for Real Value Negative Selection Algorithm BT - Soft Computing Applications and Intelligent Systems: Second International Multi-Conference on Artificial Intelligence Technology, M-CAIT 2013, Shah Alam, August 28-29, 2013. Pro', S. A. Noah, A. Abdullah, H. Arshad, A. Abu Bakar, Z. A. Othman, S. Sahran, N. Omar, and Z. Othman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 194–205.
 - [190] A. S. Perelson and G. F. Oster, 'Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination', *Journal of theoretical biology*, vol. 81, no. 4, pp. 645–670, 1979.
 - [191] E. Hart and P. Ross, 'Studies on the implications of shape-space models for idiotypic networks', in *International Conference on Artificial Immune Systems*, 2004, pp. 413–426.
 - [192] T. Stibor, J. Timmis, and C. Eckert, 'On the use of hyperspheres in artificial immune systems as antibody recognition regions', in *International Conference on Artificial Immune Systems*, 2006, pp. 215–228.
 - [193] Z. Ji and D. Dasgupta, 'V-detector: An efficient negative selection algorithm with “probably adequate” detector coverage', *Information Sciences*, vol. 179, no. 10, pp. 1390–1406, 2009, doi: <https://doi.org/10.1016/j.ins.2008.12.015>.
 - [194] M. Ayara, J. Timmis, R. de Lemos, L. N. de Castro, and R. Duncan, 'Negative Selection: How to Generate Detectors', in *1st International Conference on Artificial Immune Systems*, Sep. 2002, pp. 89–98.
 - [195] P. D'haeseleer, S. Forrest, and P. Helman, 'An immunological approach to change detection: algorithms, analysis and implications', in *Proceedings 1996 IEEE Symposium on Security and Privacy*, 1996, pp. 110–119, doi: 10.1109/SECPRI.1996.502674.
 - [196] S. T. Wierzbchoń, 'Generating optimal repertoire of antibody strings in an artificial immune system', in *Intelligent Information Systems*, Springer, 2000, pp. 119–133.
 - [197] P. Helman and S. Forrest, 'An efficient algorithm for generating random antibody strings', 1994. /paper/An-efficient-algorithm-for-generating-random-Helman-Forrest/e32e8f0c53962988daec6f3d79d6bbe181aa896e (accessed Sep. 29, 2020).
 - [198] T. Stibor, K. M. Bayarou, and C. Eckert, 'An Investigation of R-Chunk Detector Generation on Higher Alphabets BT - Genetic and Evolutionary Computation – GECCO 2004: Genetic and Evolutionary Computation Conference, Seattle, WA, USA, June 26-30, 2004. Proceedings, Part I', K. Deb, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 299–307.
 - [199] S. Singh, 'Anomaly detection using negative selection based on the r-contiguous matching rule', 2002.
 - [200] X. Hang and H. Dai, 'Constructing Detectors in Schema Complementary Space for Anomaly Detection', in *Genetic and Evolutionary Computation – GECCO 2004*, Berlin, Heidelberg, 2004, pp. 275–286, doi: 10.1007/978-3-540-24854-5_29.
 - [201] D. Li, S. Liu, and H. Zhang, 'Negative selection algorithm with constant detectors for anomaly detection', *Applied Soft Computing*, vol. 36, pp. 618–632, 2015, doi: <https://doi.org/10.1016/j.asoc.2015.08.011>.
 - [202] F. González, D. Dasgupta, and L. F. Niño, 'A Randomized Real-Valued Negative Selection Algorithm BT - Artificial Immune Systems: Second International Conference, ICARIS 2003, Edinburgh, UK,
-

- September 1-3, 2003. Proceedings', J. Timmis, P. J. Bentley, and E. Hart, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 261–272.
- [203] P. D'haeseleer, 'Further efficient algorithms for generating antibody strings', Technical Report CS95-3, The University of New Mexico, Albuquerque, NM, 1995.
- [204] J. M. Shapiro, G. B. Lamont, and G. L. Peterson, 'An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection', in Proceedings of the 7th annual conference on Genetic and evolutionary computation, New York, NY, USA, Jun. 2005, pp. 337–344, doi: 10.1145/1068009.1068063.
- [205] L. N. De Castro and F. J. Von Zuben, 'Artificial immune systems: Part II—A survey of applications', FEEC/Univ. Campinas, Campinas, Brazil, 2000.
- [206] X. Z. Gao, S. J. Ovaska, and X. Wang, 'Negative selection algorithm with applications in motor fault detection', in Soft Computing Applications in Industry, Springer, 2008, pp. 93–115.
- [207] O. Bölükbaş and H. Uğuz, 'Performance of Negative Selection Algorithms in Patient Detection and Classification', in Nature-Inspired Intelligent Techniques for Solving Biomedical Engineering Problems, IGI Global, 2018, pp. 78–102.
- [208] Z. Ji and D. Dasgupta, 'Real-Valued Negative Selection Algorithm with Variable-Sized Detectors BT - Genetic and Evolutionary Computation – GECCO 2004', 2004, pp. 287–298.
- [209] Z. Ji, 'A boundary-aware negative selection algorithm', Proceedings of the 9th International Conference on Artificial Intelligence and Soft Computing, ACTA Press, 2005.
- [210] C. Surace and K. Worden, 'A negative selection approach to novelty detection in a changing environment', Third European Workshop on Structural Health Monitoring, 2006.
- [211] C. Surace, G. Surace, and K. Worden, 'A Negative Selection Approach to detect damage in aeronautical structures with changing Operating Conditions', 26th International Congress of the Aeronautical Sciences, Anchorage, USA, 2008.
- [212] M. Li and W. X. Ren, 'Negative Selection Algorithm Using Natural Frequency for Novelty Detection under Temperature Variations', Advanced Materials Research, vol. 163–167, pp. 2747–2750, 2011, doi: 10.4028/www.scientific.net/AMR.163-167.2747.
- [213] A. Shi and X. Yu, 'Structural damage assessment using artificial immune systems and wavelet decomposition', in 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 242–247, doi: 10.1109/IJCNN.2017.7965861.
- [214] D. C. Oliveira, F. R. Chavarette, and M. L. M. Lopes, 'Damage diagnosis in an isotropic structure using an artificial immune system algorithm', Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 41, no. 11, p. 485, 2019.
- [215] M. Anaya, D. Tibaduiza, and F. Pozo, 'Data driven methodology based on artificial immune systems for damage detection', 7th European Workshop on Structural Health Monitoring, 2014.
- [216] M. Anaya, D. A. Tibaduiza, and F. Pozo, 'A bioinspired methodology based on an artificial immune system for damage detection in structural health monitoring', Shock and Vibration, vol. 2015, 2015.
- [217] B. Chen and C. Zang, 'Artificial immune pattern recognition for damage detection in structural health monitoring sensor networks', in Smart Sensor Phenomena, Technology, Networks, and Systems 2009, 2009, vol. 7293, p. 72930K.
- [218] B. Chen and C. Zang, 'Artificial immune pattern recognition for structure damage classification', Computers & Structures, vol. 87, no. 21–22, pp. 1394–1407, 2009.
- [219] B. Chen and C. Zang, 'Emergent damage pattern recognition using immune network theory', Smart Structures and Systems, vol. 8, no. 1, pp. 69–92, 2011.
- [220] B. Chen and C. Zang, 'Unsupervised structure damage classification based on the data clustering and artificial immune pattern recognition', in International Conference on Artificial Immune Systems, 2009, pp. 206–219.

-
- [221] B. Chen and C. Zang, 'A hybrid immune model for unsupervised structural damage pattern recognition', *Expert Systems with Applications*, vol. 38, no. 3, pp. 1650–1658, 2011.
- [222] A. Mojtahedi, M. A. L. Yaghin, Y. Hassanzadeh, F. Abbasidoust, M. M. Ettefagh, and M. H. Aminfar, 'A robust damage detection method developed for offshore jacket platforms using modified artificial immune system algorithm', *China Ocean Engineering*, vol. 26, no. 3, pp. 379–395, 2012.
- [223] Y. Zhou, S. Tang, X. Jia, D. Zhang, and C. Zang, 'Structural damage detection and classification algorithm based on artificial immune pattern recognition', *Journal of Shenyang Jianzhu University(Natural Science)*, vol. 29, no. 2, pp. 378–384, 2013.
- [224] Y. Zhou, S. Tang, C. Zang, and R. Zhou, 'An artificial immune pattern recognition approach for damage classification in structures', in *Advances in Information Technology and Industry Applications*, Springer, 2012, pp. 11–17.
- [225] M. G. Perhinschi, H. Moncayo, and D. Al Azzawi, 'Integrated Immunity-Based Framework for Aircraft Abnormal Conditions Management', *Journal of Aircraft*, vol. 51, no. 6, pp. 1726–1739, Oct. 2014, doi: 10.2514/1.C032381.
- [226] K. J. Nurmela and P. R. J. Östergård, 'Covering A Square With Up To 30 Equal Circles'. 2000.
- [227] A. Heppes and H. Melissen, 'Covering a Rectangle With Equal Circles', *Periodica Mathematica Hungarica*, vol. 34, no. 1, pp. 65–81, 1997, doi: 10.1023/A:1004224507766.
- [228] M. C. Markót, 'Robust Designs for Circle Coverings of a Square BT - Optimized Packings with Applications', G. Fasano and J. D. Pintér, Eds. Cham: Springer International Publishing, 2015, pp. 225–242.
- [229] F. A. González and D. Dasgupta, 'Anomaly Detection Using Real-Valued Negative Selection', *Genetic Programming and Evolvable Machines*, vol. 4, no. 4, pp. 383–403, 2003, doi: 10.1023/A:1026195112518.
- [230] S. A. Hofmeyr and S. Forrest, 'Architecture for an Artificial Immune System', *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, Dec. 2000, doi: 10.1162/106365600568257.
- [231] A. K. Pandey, M. Biswas, and M. M. Samman, 'Damage detection from changes in curvature mode shapes', *Journal of sound and vibration*, vol. 145, no. 2, pp. 321–332, 1991.
- [232] A. K. Pandey and M. Biswas, 'Damage detection in structures using changes in flexibility', *Journal of sound and vibration*, vol. 169, no. 1, pp. 3–17, 1994.
- [233] P. Cawley and R. D. Adams, 'The location of defects in structures from measurements of natural frequencies', *The Journal of Strain Analysis for Engineering Design*, vol. 14, no. 2, pp. 49–57, Apr. 1979, doi: 10.1243/03093247V142049.
- [234] R. D. Adams, P. Cawley, C. J. Pye, and B. J. Stone, 'A vibration technique for non-destructively assessing the integrity of structures', *Journal of Mechanical Engineering Science*, vol. 20, no. 2, pp. 93–100, 1978.
- [235] N. Stubbs, J.-T. Kim, and C. R. Farrar, 'Field verification of a nondestructive damage localization and severity estimation algorithm', in *Proceedings-SPIE the international society for optical engineering*, 1995, p. 210.
- [236] Z. Zhang and A. E. Aktan, 'The damage indices for the constructed facilities', in *Proceedings-spie the International Society for Optical Engineering*, 1995, p. 1520.
- [237] D. C. Zimmerman and M. Kaouk, 'Structural damage detection using a minimum rank update theory', 1994.
- [238] T. Wolff and M. Richardson, 'Fault detection in structures from changes in their modal parameters', in *Proceedings of the 7th international modal analysis conference*, 1989, vol. 1, pp. 87–94.
- [239] G. Manson, K. Worden, and D. Allman, 'Experimental Validation of a Structural Health Monitoring Methodology: Part III. Damage Location on an Aircraft Wing', *Journal of Sound and Vibration*, vol. 259, no. 2, pp. 365–385, 2003, doi: <https://doi.org/10.1006/jsvi.2002.5169>.
-

- [240] S. Nagarajaiah, B. Basu, and Y. Yang, '4 - Output only modal identification and structural damage detection using timefrequency and wavelet techniques for assessing and monitoring civil infrastructures', in Woodhead Publishing Series in Electronic and Optical Materials, vol. 56, M. L. Wang, J. P. Lynch, and H. B. T.-S. T. for C. I. Sohn, Eds. Woodhead Publishing, 2014, pp. 93–144.
- [241] M. L. Fugate, H. Sohn, and C. R. Farrar, 'Vibration-based damage detection using statistical process control', *Mechanical Systems and Signal Processing*, vol. 15, no. 4, pp. 707–721, 2001.
- [242] M. Gul and F. N. Catbas, 'Statistical pattern recognition for Structural Health Monitoring using time series modeling: Theory and experimental verifications', *Mechanical Systems and Signal Processing*, vol. 23, no. 7, pp. 2192–2204, 2009.
- [243] D. J. Ewins, *Modal testing: theory and practice*, vol. 15. Research studies press Letchworth, 1984.
- [244] N. M. M. Maia and J. M. M. Silva, 'Modal analysis identification techniques', *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 359, no. 1778, pp. 29–40, 2001.
- [245] L. F. Ramos, 'Damage identification on masonry structures based on vibration signatures', PhD thesis, University of Minho, 2007.
- [246] S. S. Ivanović, M. D. Trifunac, E. I. Novikova, A. A. Gladkov, and M. I. Todorovska, 'Ambient vibration tests of a seven-story reinforced concrete building in Van Nuys, California, damaged by the 1994 Northridge earthquake', *Soil Dynamics and Earthquake Engineering*, vol. 19, no. 6, pp. 391–411, 2000.
- [247] B. Peeters and C. E. Ventura, 'Comparative study of modal analysis techniques for bridge dynamic characteristics', *Mechanical Systems and Signal Processing*, vol. 17, no. 5, pp. 965–988, 2003.
- [248] Á. Cunha, E. Caetano, R. Brincker, and P. Andersen, 'Identification from the natural response of Vasco da Gama bridge', 2004.
- [249] P. Andersen, R. Brincker, B. Peeters, G. De Roeck, L. Hermans, and C. Krämer, 'Comparison of system identification methods using ambient bridge test data', in *Proc. of the 17th international modal analysis conference*, Kissimmee, Florida, 1999, pp. 7–10.
- [250] T. Horyna and C. E. Ventura, 'Summary of HCT building ambient vibration data analyses', in *SPIE proceedings series*, 2000, pp. 1095–1098.
- [251] W. Guo, Y. Chen, T. Wang, and H. Tian, 'Optimization of Real-Valued Self Set in Immunity-based WSN Intrusion Detection BT - 2016 International Conference on Artificial Intelligence and Engineering Applications', Nov. 2016, doi: <https://doi.org/10.2991/aiea-16.2016.22>.
- [252] L. Xi, F. Zhang, and D. Wang, 'Optimization of Real-Valued Self Set for Anomaly Detection Using Gaussian Distribution BT - Artificial Intelligence and Computational Intelligence', 2009, pp. 112–120.
- [253] J. Zhang and W. Luo, 'On the self representation based on the adaptive self radius', in *The Fourth International Workshop on Advanced Computational Intelligence*, 2011, pp. 157–163, doi: 10.1109/IWACI.2011.6159993.
- [254] D. C. Montgomery, *Design and Analysis of Experiments*. John Wiley & Sons, 2008.
- [255] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. R. Stewart, 'Designing and reporting on computational experiments with heuristic methods', *Journal of Heuristics*, vol. 1, no. 1, pp. 9–32, 1995, doi: 10.1007/BF02430363.
- [256] R. L. Rardin and R. Uzsoy, 'Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial', *Journal of Heuristics*, vol. 7, no. 3, pp. 261–304, 2001, doi: 10.1023/A:1011319115230.
- [257] C. C. McGeoch, 'Feature Article—Toward an Experimental Method for Algorithm Simulation', *INFORMS Journal on Computing*, vol. 8, no. 1, pp. 1–15, Feb. 1996, doi: 10.1287/ijoc.8.1.1.
- [258] S. L. Salzberg, 'On Comparing Classifiers: A Critique of Current Research and Methods', 1999.

-
- [259] B. Adenso-Díaz and M. Laguna, 'Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search', *Operations Research*, vol. 54, no. 1, pp. 99–114, Feb. 2006, doi: 10.1287/opre.1050.0243.
 - [260] K. Sörensen, 'Metaheuristics—the metaphor exposed', *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.
 - [261] E.-G. Talbi, *Metaheuristics: from design to implementation*, vol. 74. John Wiley & Sons, 2009.
 - [262] D. H. Wolpert, 'The Lack of A Priori Distinctions Between Learning Algorithms', *Neural Computation*, vol. 8, pp. 1341–1390, 1996.
 - [263] D. H. Wolpert and W. G. Macready, 'No free lunch theorems for optimization', *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997, doi: 10.1109/4235.585893.
 - [264] X.-S. Yang, 'Swarm-based metaheuristic algorithms and no-free-lunch theorems', in *Theory and New Applications of Swarm Intelligence*, InTech, 2012.
 - [265] M. Sokolova and G. Lapalme, 'A systematic analysis of performance measures for classification tasks', *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009, doi: <https://doi.org/10.1016/j.ipm.2009.03.002>.
 - [266] W. J. Youden, 'Index for rating diagnostic tests', *Cancer*, vol. 3, no. 1, pp. 32–35, Jan. 1950, doi: 10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.
 - [267] R. Caruana and A. Niculescu-Mizil, 'Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria', in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 69–78, doi: 10.1145/1014052.1014063.
 - [268] T. Fawcett, 'ROC Graphs: Notes and Practical Considerations for Researchers', 2004.
 - [269] T. Saito and M. Rehmsmeier, 'The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets', *PLOS ONE*, vol. 10, no. 3, pp. 1–21, 2015, doi: 10.1371/journal.pone.0118432.
 - [270] J. Huang and C. X. Ling, 'Constructing New and Better Evaluation Measures for Machine Learning', in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 859–864.
 - [271] T. G. Dietterich, 'Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms', *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998, doi: 10.1162/089976698300017197.
 - [272] H. P. Crowder, R. S. Dembo, and J. M. Mulvey, 'Reporting computational experiments in mathematical programming', *Mathematical Programming*, vol. 15, no. 1, pp. 316–329, 1978, doi: 10.1007/BF01609036.
 - [273] J. JCSS, 'Probabilistic model code', *Joint Committee on Structural Safety*, 2001.
 - [274] Fédération Internationale du Béton, 'Model Code 2010—First Complete Draft', *fib Bulletin*, no. 55, p. 318, 2010.
 - [275] J. Kittler, 'Combining classifiers: A theoretical framework', *Pattern Anal. Appl.*, vol. 1, no. 1, pp. 18–27, Mar. 1998, doi: 10.1007/BF01238023.
 - [276] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and Josef Kittler, 'Combining multiple classifiers by averaging or by multiplying?', *Pattern Recognition*, vol. 33, no. 9, pp. 1475–1485, Sep. 2000, doi: 10.1016/S0031-3203(99)00138-7.
 - [277] C. R. Farrar, W. E. Baker, T. M. Bell, K. M. Cone, T. W. Darling, T. A. Duffey, A. Eklund, and A. Migliori, 'Dynamic characterization and damage detection in the I-40 bridge over the Rio Grande', United States, 1994. doi: 10.2172/10158042.
 - [278] S.-E. Fang and R. Perera, 'Power mode shapes for early damage detection in linear structures', *Journal of Sound and Vibration*, vol. 324, no. 1, pp. 40–56, 2009, doi: <https://doi.org/10.1016/j.jsv.2009.02.002>.
-

- [279] ANSYS, User's manual, revision 8.0. Canonsburg, Pennsylvania: Ansys Inc., 2003.
- [280] C. R. Farrar, T. A. Duffey, P. A. Goldman, D. V. Jauregui, and J. S. Vigil, 'Finite element analysis of the I-40 bridge over the Rio Grande', Los Alamos National Lab., NM (United States), 1996.
- [281] A. Barontini, R. Perera, M. G. Masciotta, P. Amado-Mendes, L. Ramos, and P. Lourenço, 'Deterministically generated negative selection algorithm for damage detection in civil engineering systems', *Engineering Structures*, vol. 197, pp. 109444–109444, 2019, doi: 10.1016/j.engstruct.2019.109444.
- [282] M. M. Abdel Wahab and G. De roeck, 'Damage Detection in Bridges Using Modal Curvatures: Application to a Real Damage Scenario', *Journal of Sound and Vibration*, vol. 226, no. 2, pp. 217–235, Sep. 1999, doi: 10.1006/jsvi.1999.2295.
- [283] F. Magalhães, A. Cunha, and E. Caetano, 'Vibration based structural health monitoring of an arch bridge: From automated OMA to damage detection', *Mechanical Systems and Signal Processing*, vol. 28, pp. 212–228, Apr. 2012, doi: 10.1016/j.ymssp.2011.06.011.
- [284] B. S. Wang and Z. C. He, 'Crack detection of arch dam using statistical neural network based on the reductions of natural frequencies', *Journal of Sound and Vibration*, vol. 302, no. 4, pp. 1037–1047, May 2007, doi: 10.1016/j.jsv.2007.01.008.
- [285] M. I. Todorovska and M. D. Trifunac, 'Earthquake damage detection in the Imperial County Services Building III: Analysis of wave travel times via impulse response functions', *Soil Dynamics and Earthquake Engineering*, vol. 28, no. 5, pp. 387–404, May 2008, doi: 10.1016/j.soildyn.2007.07.001.
- [286] A. A. Elshafey, M. R. Haddara, and H. Marzouk, 'Damage detection in offshore structures using neural networks', *Marine Structures*, vol. 23, no. 1, pp. 131–145, Jan. 2010, doi: 10.1016/j.marstruc.2010.01.005.
- [287] M. Rahmani, M. Ebrahimian, and M. I. Todorovska, 'Time-wave velocity analysis for early earthquake damage detection in buildings: application to a damaged full-scale RC building', *Earthquake Engineering & Structural Dynamics*, vol. 44, no. 4, pp. 619–636, 2015, doi: <https://doi.org/10.1002/eqe.2539>.
- [288] H. Li, J. Wang, and S.-L. James Hu, 'Using incomplete modal data for damage detection in offshore jacket structures', *Ocean Engineering*, vol. 35, no. 17, pp. 1793–1799, Dec. 2008, doi: 10.1016/j.oceaneng.2008.08.020.
- [289] P. B. Lourenço, K. J. Krakowiak, F. M. Fernandes, and L. F. Ramos, 'Failure analysis of Monastery of Jerónimos, Lisbon: How to learn from sophisticated numerical models', *Engineering Failure Analysis*, vol. 14, no. 2, pp. 280–300, 2007, doi: <https://doi.org/10.1016/j.engfailanal.2006.02.002>.
- [290] A. Barontini, M.-G. Masciotta, L. F. Ramos, P. Amado-Mendes, and P. B. Lourenço, 'Application of a Bio-Inspired Anomaly Detection Algorithm for Unsupervised SHM of a Historic Masonry Church', in *Proceedings of the Tenth International Masonry Conference*, 2018, pp. 2259–2272.
- [291] A. Barontini, M. G. Masciotta, P. A. Mendes, and L. F. Ramos, 'Performance assessment of a bio-inspired anomaly detection algorithm for unsupervised SHM: application to a Manueline masonry church', *IJMRI*, vol. 5, no. 4, p. 468, 2020, doi: 10.1504/IJMRI.2020.111798.
- [292] M. G. Masciotta, D. Pellegrini, D. Brigante, A. Barontini, P. B. Lourenço, M. Girardi, C. Padovani, and G. Fabbrocino, 'Dynamic characterization of progressively damaged segmental masonry arches with one settled support: experimental and numerical analyses', *Frattura ed Integrità Strutturale*, vol. 14, no. 51 SE-SI: Fracture and Damage Detection in Masonry Structures, pp. 423–441, Dec. 2019, doi: 10.3221/IGF-ESIS.51.31.
- [293] M.-G. Masciotta, D. Brigante, A. Barontini, D. Pellegrini, P. B. Lourenço, G. Fabbrocino, C. Padovani, M. Girardi, A. Arede, and C. Costa, 'Experimental and Numerical Investigations of a

- Segmental Masonry Arch Subjected to Horizontal Settlements BT - Proceedings of ARCH 2019', 2020, pp. 413–421.
- [294] 'ARTEMIS Modal S 3.1.1'. 2018.
- [295] A. Barontini, M. G. Masciotta, P. Amado-Mendes, L. F. Ramos, and P. B. Lourenço, 'Application of a classification algorithm to the early-stage damage detection of a masonry arch', presented at the XI International Conference on Structural Dynamics (EURODYN), Athens, Greece, Nov. 2020, doi: 10.47964/1120.9105.19704.
- [296] A. Barontini, M. G. Masciotta, P. Amado-Mendes, L. F. Ramos, and P. B. Lourenço, 'Negative selection algorithm based methodology for online structural health monitoring', *Engineering Structures*, vol. 229, p. 111662, Feb. 2021, doi: 10.1016/j.engstruct.2020.111662.
- [297] S. Gres, M. D. Ulriksen, M. Dolher, R. Johansen, P. Andersen, L. Damkilde, and S. A. Nielsen, 'Statistical methods for damage detection applied to civil structures', *Procedia Engineering*, vol. 199, pp. 1919–1924, 2017, doi: 10.1016/j.proeng.2017.09.280.
- [298] H. Sohn, C. R. Farrar, N. F. Hunter, and K. Worden, 'Structural Health Monitoring Using Statistical Pattern Recognition Techniques', *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 4, pp. 706–711, Feb. 2001, doi: 10.1115/1.1410933.
- [299] C. Hsu, C. Chang, and C. Lin, 'A practical guide to support vector classification'. .
- [300] C. J. C. Burges, 'A Tutorial on Support Vector Machines for Pattern Recognition', *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998, doi: 10.1023/A:1009715923555.

APPENDIX

Table A1: 81 combinations in each test carried out.

CASE	CensDIST	Div	r det	r self	t	CASE	CensDIST	Div	r det	r self	t
1	0.0101	70	0.0101	0.01	15	42	0.0251	35	0.0202	0.03	45
2	0.0101	70	0.0101	0.01	30	43	0.0451	35	0.0202	0.07	15
3	0.0101	70	0.0101	0.01	45	44	0.0451	35	0.0202	0.07	30
4	0.0101	70	0.0101	0.03	15	45	0.0451	35	0.0202	0.07	45
5	0.0101	70	0.0101	0.03	30	46	0.0404	10	0.0707	0.01	15
6	0.0101	70	0.0101	0.03	45	47	0.0404	10	0.0707	0.01	30
7	0.0101	70	0.0101	0.07	15	48	0.0404	10	0.0707	0.01	45
8	0.0101	70	0.0101	0.07	30	49	0.0504	10	0.0707	0.03	15
9	0.0101	70	0.0101	0.07	45	50	0.0504	10	0.0707	0.03	30
10	0.0202	35	0.0202	0.01	15	51	0.0504	10	0.0707	0.03	45
11	0.0202	35	0.0202	0.01	30	52	0.0704	10	0.0707	0.07	15
12	0.0202	35	0.0202	0.01	45	53	0.0704	10	0.0707	0.07	30
13	0.0202	35	0.0202	0.03	15	54	0.0704	10	0.0707	0.07	45
14	0.0202	35	0.0202	0.03	30	55	0.0201	70	0.0101	0.01	15
15	0.0202	35	0.0202	0.03	45	56	0.0201	70	0.0101	0.01	30
16	0.0202	35	0.0202	0.07	15	57	0.0201	70	0.0101	0.01	45
17	0.0202	35	0.0202	0.07	30	58	0.0401	70	0.0101	0.03	15
18	0.0202	35	0.0202	0.07	45	59	0.0401	70	0.0101	0.03	30
19	0.0707	10	0.0707	0.01	15	60	0.0401	70	0.0101	0.03	45
20	0.0707	10	0.0707	0.01	30	61	0.0801	70	0.0101	0.07	15
21	0.0707	10	0.0707	0.01	45	62	0.0801	70	0.0101	0.07	30
22	0.0707	10	0.0707	0.03	15	63	0.0801	70	0.0101	0.07	45
23	0.0707	10	0.0707	0.03	30	64	0.0302	35	0.0202	0.01	15
24	0.0707	10	0.0707	0.03	45	65	0.0302	35	0.0202	0.01	30
25	0.0707	10	0.0707	0.07	15	66	0.0302	35	0.0202	0.01	45
26	0.0707	10	0.0707	0.07	30	67	0.0502	35	0.0202	0.03	15
27	0.0707	10	0.0707	0.07	45	68	0.0502	35	0.0202	0.03	30
28	0.0101	70	0.0101	0.01	15	69	0.0502	35	0.0202	0.03	45
29	0.0101	70	0.0101	0.01	30	70	0.0902	35	0.0202	0.07	15
30	0.0101	70	0.0101	0.01	45	71	0.0902	35	0.0202	0.07	30
31	0.0201	70	0.0101	0.03	15	72	0.0902	35	0.0202	0.07	45
32	0.0201	70	0.0101	0.03	30	73	0.0807	10	0.0707	0.01	15
33	0.0201	70	0.0101	0.03	45	74	0.0807	10	0.0707	0.01	30
34	0.0401	70	0.0101	0.07	15	75	0.0807	10	0.0707	0.01	45
35	0.0401	70	0.0101	0.07	30	76	0.1007	10	0.0707	0.03	15
36	0.0401	70	0.0101	0.07	45	77	0.1007	10	0.0707	0.03	30
37	0.0151	35	0.0202	0.01	15	78	0.1007	10	0.0707	0.03	45
38	0.0151	35	0.0202	0.01	30	79	0.1407	10	0.0707	0.07	15
39	0.0151	35	0.0202	0.01	45	80	0.1407	10	0.0707	0.07	30
40	0.0251	35	0.0202	0.03	15	81	0.1407	10	0.0707	0.07	45
41	0.0251	35	0.0202	0.03	30						

Table A2: Coefficients of the response surface models.

Instance	dist level	β_0	β_1	β_2	β_3	β_4	β_5
T1.1	-1	0.437	-55.806	-1.703E-13	8.961	3.066E-14	-4.081E-13
T1.1	0	0.418	-20.409	-16.104	3.363	6.419	-25.276
T1.1	1	0.457	-46.919	-79.971	7.629	11.414	-71.205
T1.2	-1	0.528	-77.145	-1.459E-13	10.323	2.445E-14	-3.439E-13
T1.2	0	0.512	2.449	-75.849	2.056	10.796	-37.535
T1.2	1	0.772	-7.639	-48.269	1.635	4.387	-35.919
T1.3	-1	0.656	-60.810	-3.985E-14	7.472	1.057E-14	-2.072E-13
T1.3	0	0.682	-7.196	-52.023	1.335	6.845	-19.987
T1.3	1	0.899	-6.237	-23.301	-0.157	0.498	-7.832
T2.1	-1	0.423	-58.116	-3.224	9.105	0.266	0.420
T2.1	0	0.439	-3.514	-12.176	1.328	6.061	-10.399
T2.1	1	0.439	-59.078	-97.025	8.478	12.876	-85.805
T2.2	-1	0.491	-139.028	1.287	14.847	-0.082	1.790
T2.2	0	0.471	-25.094	-92.859	3.551	12.349	-34.019
T2.2	1	0.740	-51.671	-61.362	5.018	5.666	-49.578
T2.3	-1	0.657	-102.886	7.670	10.488	-0.608	0.927
T2.3	0	0.661	-46.341	-79.638	2.935	8.393	0.196
T2.3	1	0.877	-28.224	-29.227	2.047	1.391	-28.534

Table A3: Combination of levels for the Full Factorial Design (FFD) and the Central Composite Design (CCD).

Set of instances	FFD			CCD		
Combination	r_{det}	r_{self}	dist	r_{det}	r_{self}	dist
1	0.0101	0.0101	0.0101	0.0139	0.0139	0.0139
2	0.0101	0.0101	0.0101	0.0139	0.0139	0.0139
3	0.0101	0.0101	0.0202	0.0139	0.0139	0.0277
4	0.0101	0.0354	0.0101	0.0544	0.0139	0.0544
5	0.0101	0.0354	0.0227	0.0544	0.0139	0.0341
6	0.0101	0.0354	0.0455	0.0544	0.0139	0.0683
7	0.0101	0.0707	0.0101	0.0139	0.0544	0.0139
8	0.0101	0.0707	0.0404	0.0139	0.0544	0.0341
9	0.0101	0.0707	0.0808	0.0139	0.0544	0.0683
10	0.0354	0.0101	0.0202	0.0544	0.0544	0.0544
11	0.0354	0.0101	0.0152	0.0544	0.0544	0.0544
12	0.0354	0.0101	0.0303	0.0544	0.0544	0.1088
13	0.0354	0.0354	0.0202	0.0354	0.0354	0.0354
14	0.0354	0.0354	0.0278	0.0354	0.0354	0.0354
15	0.0354	0.0354	0.0556	0.0354	0.0354	0.0707
16	0.0354	0.0707	0.0202	0.0101	0.0354	0.0101
17	0.0354	0.0707	0.0455	0.0101	0.0354	0.0227
18	0.0354	0.0707	0.0909	0.0101	0.0354	0.0455
19	0.0707	0.0101	0.0707	0.0707	0.0354	0.0707
20	0.0707	0.0101	0.0404	0.0707	0.0354	0.0530
21	0.0707	0.0101	0.0808	0.0707	0.0354	0.1061
22	0.0707	0.0354	0.0707	0.0354	0.0101	0.0354
23	0.0707	0.0354	0.0530	0.0354	0.0101	0.0227
24	0.0707	0.0354	0.1061	0.0354	0.0101	0.0455
25	0.0707	0.0707	0.0707	0.0354	0.0707	0.0354
26	0.0707	0.0707	0.0707	0.0354	0.0707	0.0530
27	0.0707	0.0707	0.1414	0.0354	0.0707	0.1061

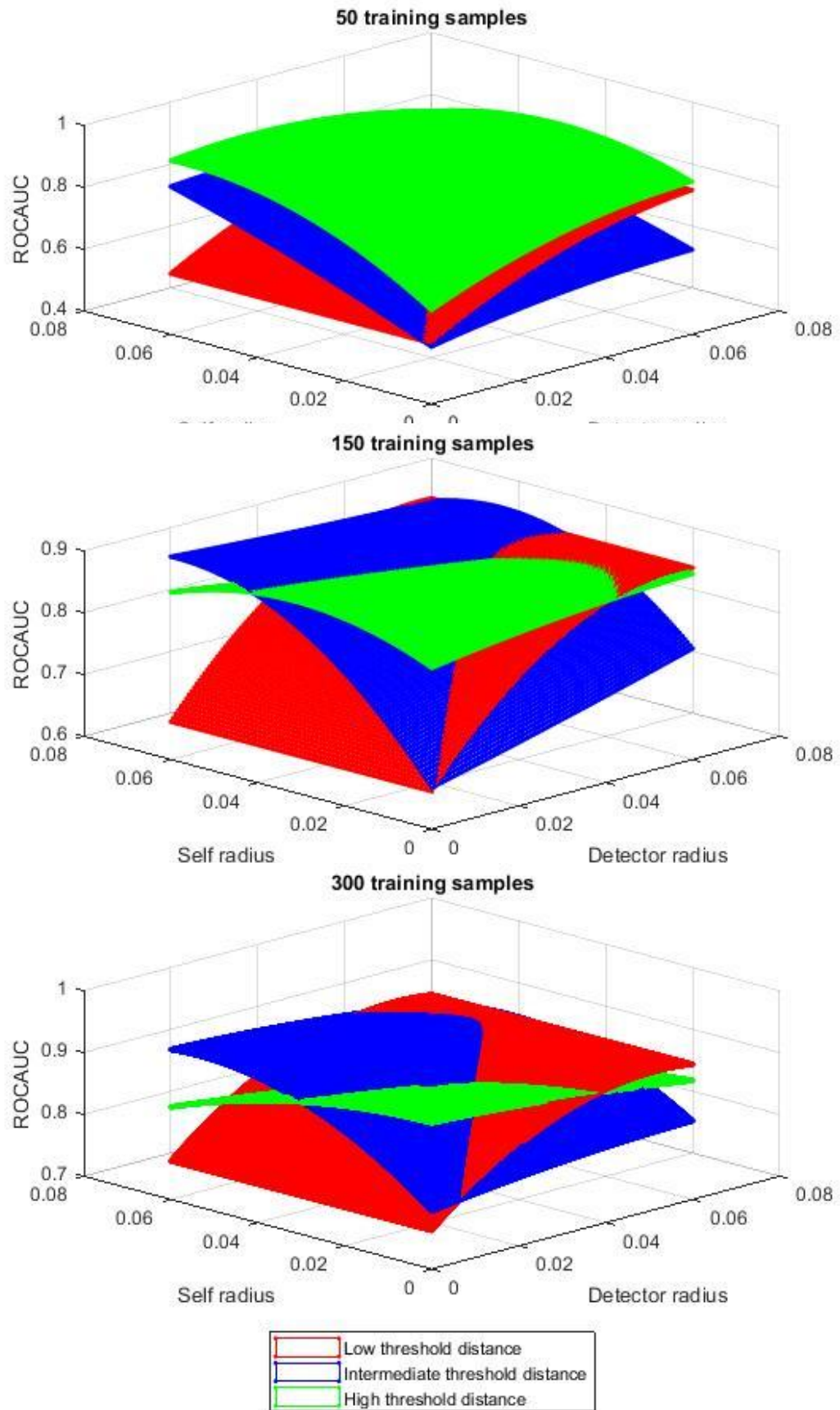


Figure A1: Surface plot of the regression model for the three problem instances and the three censoring distances, full factorial design.

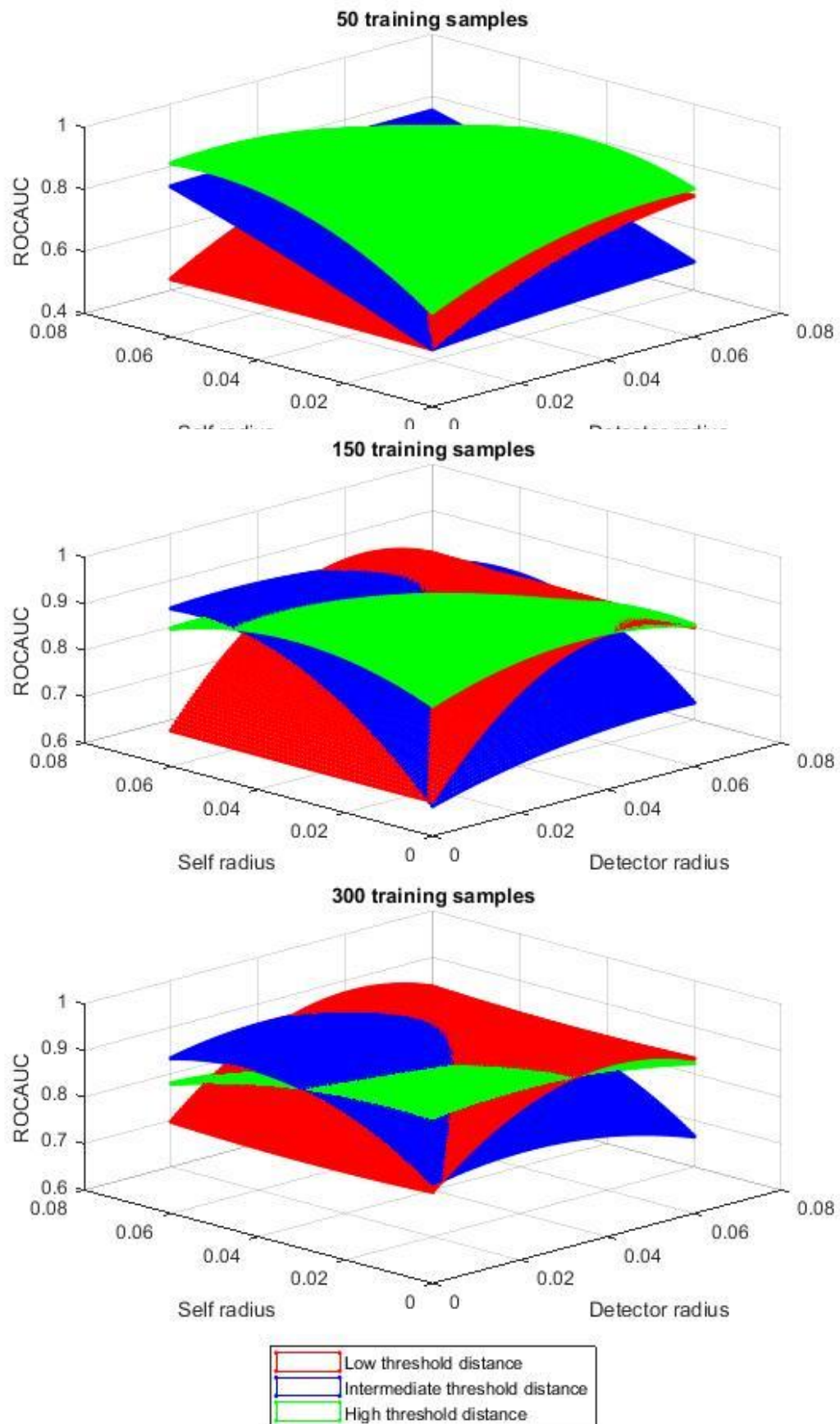


Figure A2: Surface plot of the regression model for the three problem instances and the three censoring distances, central composite design.