Pontifícia Universidade Católica
do Rio de Janeiro

Mateus Gheorghe de Castro Ribeiro

Guided waves-based structural damage evaluation with machine learning

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica.

Advisor : Prof. Helon Vicente Hultmann Ayala
Co-advisor: Prof. Alan Conci Kubrusly

Rio de Janeiro
December 2020

Pontifícia Universidade Católica
do Rio de Janeiro

Mateus Gheorghe de Castro Ribeiro

Guided waves-based structural damage evaluation with machine learning

Dissertation presented to the Programa de Pós–graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica. Approved by the Examination Committee:

Prof. Helon Vicente Hultmann Ayala
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Alan Conci Kubrusly
Co-advisor
Centro de Estudos em Telecomunicações (CETUC) – PUC-Rio

Prof. Arthur Martins Barbosa Braga
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Igor Braga de Paula
Departamento de Engenharia Mecânica – PUC-Rio

Rio de Janeiro, December the 9th, 2020

Mateus Gheorghe de Castro Ribeiro

Majored in mechanical engineering by the Federal University of Juiz de Fora (Juiz de Fora-MG, Brazil) in 2018.

Bibliographic data

Ribeiro, Mateus Gheorghe De Castro

Guided waves-based structural damage evaluation with machine learning / Mateus Gheorghe de Castro Ribeiro; advisor: Helon Vicente Hultmann Ayala; co-advisor: Alan Conci Kubrusly. – 2020.

87 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2020.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Monitoramento de saúde estrutural. 3. Ondas guiadas. 4. Aprendizado de máquina. 5. Conversão de modo. 6. Aprendizado supervisionado. I. Ayala, Helon Vicente Hultmann. II. Kubrusly, Alan Conci. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. IV. Título.

CDD: 621

To my parents, for their support
and encouragement.

Acknowledgments

It is a pleasure to recognize and thank those who made this dissertation possible. All of those that I cite contribute in different ways to the conclusion of this document.

I would like to firstly thank god for protection and assistance in every choice made.

I have certainty that this document wouldn't be accomplished without my supervisors' aid. I owe my deepest gratitude to my supervisor, Professor D.Sc. Helon Ayala, for continuous guidance and support, for encouragement in scientific research and contributions, for the opportunities provided, as well as the professional and academic advice. I also express my profound gratitude to my Co-advisor, Professor D.Sc. Alan Kubrusly, whose feedback and support were of extreme importance. Thank you both for your patience and trust.

I am deeply grateful to Professor D.Sc. Arthur Braga, for the opportunity to work in one of the most promising projects I ever imagined working with, and, also, in one of the greatest laboratories in PUC-Rio.

I would like to thank all the continuous support provided by my parents, Claudete and Gilnei, who have always been by my side and were essential in the construction of the foundations of my life. Needless to say, I would like to thank my beloved Leticia, who always encourage me to follow my dreams and, sometimes, even believes in me more than I do. Thanks for your patience and comprehension.

I would like to thank my friend Renan Finotti for the encouragement and incentive to pursue my master's degree in PUC-Rio. It is also important to acknowledge, Pedro Calderano and Guilherme Bessa, undergraduate friends which followed the same path as I did, thanks for the friendship, you made life far from home easier. I cannot forget the LSFO team: Juan Hidalgo, Luis Brasil, Isabel Camerini, Tiago Magalhães, Louise Erthal, and Thiago Cavalcante, just to name a few, for the collaboration and fellowship.

In addition, I thank the reviewers of my dissertation, once again Professor D.Sc. Arthur Braga, and Professor D.Sc. Igor Braga de Paula for the detailed reviews and comments of extreme importance. Those were critical to the improvement and completion of this work.

Finally, I thank CNPq and CAPES for all the financial aid provided. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) - Finance Code 001.

Abstract

Ribeiro, Mateus Gheorghe De Castro; Ayala, Helon Vicente Hultmann (Advisor); Kubrusly, Alan Conci (Co-Advisor). **Guided waves-based structural damage evaluation with machine learning**. Rio de Janeiro, 2020. 87p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Recently ultrasonic guided waves have shown great potential for nondestructive testing and structural health monitoring (SHM) in a damage evaluation scenario. Measurements utilizing elastic waves are particularly useful due to their capability to propagate in different materials such as solid and fluid bounded media, and, also, the ability to cover broad areas. When enough guided waves measurements are available and advanced data-driven techniques such as machine learning can be applied to the problem, the damage evaluation procedure becomes then even more powerful and robust. Based on these circumstances, the present work deals with the application of machine learning models to provide fault evaluation inferences based on ultrasonic guided waves information. Two main case studies are tackled in the mentioned subject. Firstly, a carbon fiber reinforced polymer (CFRP) plate is assessed using open data of Lamb guided wave signals in the detection of dot type defects. Results demonstrated that a baseline dependent approach can obtain excellent results when using system identification feature extraction. Secondly, corrosion-like defects in an aluminium plate are classified according to their severity. The methodology is assisted by a mode separation scheme of SH guided waves signals of pre-acquired data. Results have shown that the adoption of mode separation can in fact improve the machine learning results.

Keywords

Structural health monitoring; Guided-waves; Machine learning; Mode conversion; Supervised learning.

Resumo

Ribeiro, Mateus Gheorghe De Castro; Ayala, Helon Vicente Hultmann; Kubrusly, Alan Conci. **Avaliação de danos estruturais baseada em ondas guiadas ultrassônicas e aprendizado de máquina**. Rio de Janeiro, 2020. 87p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Recentemente, ondas guiadas por ultrassom têm mostrado grande potencial para ensaios não destrutivos e monitoramento de integridade estrutural (SHM) em um cenário de avaliação de danos. As medições obtidas por meio de ondas elásticas são particularmente úteis devido a sua capacidade de se propagarem em diferentes materiais, como meios sólidos e fluidos e, também, a capacidade de abrangerem áreas amplas. Ao possuir suficientes medições oriundas de ondas guiadas, técnicas avançadas baseadas em dados, como aprendizado de máquina, podem ser aplicadas ao problema, tornando o procedimento de avaliação de danos ainda mais poderoso e robusto. Com base nessas circunstâncias, o presente trabalho trata da aplicação de modelos de aprendizado de máquina para fornecer inferências de avaliação de falhas baseadas em informações de ondas guiadas por ultrassom. Dois principais estudos de caso são abordados. Primeiramente, uma placa de polímero reforçado com fibra de carbono (PRFC) é avaliada, utilizando dados da literatura de sinais de onda guiada do tipo Lamb na detecção de defeitos pontuais. Os resultados demonstraram que uma abordagem que utiliza um sinal de referência foi capaz de obter excelentes acurácias ao usar a extração de características baseadas em técnicas de identificação de sistemas. Em um segundo momento, defeitos semelhantes à corrosão em uma placa de alumínio são classificados de acordo com sua gravidade. A metodologia é auxiliada por um esquema de separação de modos em sinais de ondas guiadas do tipo *SH* pré-adquiridos. Os resultados obtidos mostraram que a adoção da separação de modos pode, de fato, melhorar os resultados do aprendizado de máquina.

Palavras-chave

Monitoramento de saúde estrutural; Ondas guiadas; Aprendizado de máquina; Conversão de modo; Aprendizado supervisionado.

Table of contents

List of Figures

List of Tables

List of Abreviations

Adam – Adaptive Moment Estimation

AI – Artificial Inteligence

ANN – Artificial Neural Network

AR – Auto-Regressive Models

AUAC – Area Under the Accuracy Curve

CE – Categorical Cross-Entropy

CFRP – Carbon Fiber Reinforced Polymer

EC – Evolutionary Computation

FEM – Finite Element Method

FN – False Negative

FP – False Positive

GW – Guided Waves

LM – Linear Models

LOOCV – Leave-One-Out Cross-Validation

ML – Machine Learning

MLP – Multilayer Perceptron

NAR – Non-Linear Auto-Regressive Models

NARMA – Non-linear Auto-Regressive Moving Average Models

PPM EMAT – Periodic Permanent Magnet Electromagnetic Acoustic Transducers

ReLU – Rectified Linear Unit

RH – Relative Humidity

RMSE – Root Mean Squared Error

SCG – Scaled Conjugate Gradient

SGD – Stochastic Gradient Descent

SH – Shear Horizontal

SHM – Structural Health Monitoring

TN – True Negative

TP – True Positive

You never regret taking risks, but you will regret not doing the thing that was risky

Yann LeCun, .

1 Introduction

In the current century, Artificial Intelligence (AI) has been receiving growing attention. It was created long before and was defined by its pioneer John McCartney as “the science and engineering of making intelligent machines” [1]. It can also be defined as a universal field where a combination of mathematics and engineering reflects human behavior [2]. In fact, the various areas of AI are inspired by the human being, based either on their behavior, biology, or nature. Many are the foundations of AI such as fuzzy logic systems which reflect the reasoning of human in the sense that things are not binary as boolean logic [3]; artificial neural networks (ANN), which are inspired by the human biological systems, more specifically on the neurons and their connections [4]; and, also, evolutionary computation which are based on the darwinian theory and natural selection [5].

An important and resourceful branch of AI uses algorithms which learn from example data or past experience. It is then able to make inferences in unseen data, the procedure or subset of study is the so-called Machine Learning (ML) [6]. The techniques are able to solve complex problems and identify complicated patterns as long as sufficient data is available. Consequently, the applications using ML are spanning in diversified areas such as image recognition [7, 8], medical diagnosis [9, 10] and stock market trading [11, 12] extending to self driving cars [13, 14], language translation [15, 16] and structural health monitoring [17, 18].

Structural Health Monitoring (SHM) is one of the main damage detection strategies [19]. The concept of SHM is to record the response of a structure to a diagnostic input and analyze the response through a physical model or data-driven statistical and ML techniques [20]. One of the most popular physical signal measurements for the SHM model is the guided wave (GW), since it can travel long distances and propagate into different media and geometries [21]. There are different types of GW that can travel in a handful of structures and materials enabling the evaluation of distinguishing types of damages or faults in places that are not reachable by the operator and/or highly costly to do so.

1.1 Structural Health Monitoring

Structural Health Monitoring (SHM) based on GW has shown great potential in many applications either with Lamb waves [22, 23, 24, 25, 26] and, also with Shear Horizontal (SH) GW [27, 28, 29, 30, 31]. SH GW are widely used due to their characteristics such as simple dispersion relations and modes' profiles, as well as not being affected by a non-viscous liquid in contact with the waveguide's surfaces, being thus especially attractive for inspection of pipes [27, 30]. Depending on the material, geometry and frequency, several SH GW modes are able to propagate [32].

Lamb waves are also an option concerning SHM. They can be attractive in inspections of large structures, offering coverage of tens of meters [33] and, can be useful when using modes in non-dispersive regions [34]. Another positive point of Lamb waves is the capacity of traveling in materials with high attenuation ratios such as carbon fiber reinforced composites (CFRP). Additionally, it is able to detect internal and superficial defects [24]. Furthermore, in comparison with SH waves, Lamb waves are easily excited with various configurations of transducers which may be the reason for its popularity while, on the other hand, the former are less popular and dependent on specific types of transducers (i.e. periodic permanent magnet electromagnetic acoustic transducers - PPM EMAT). However, the propagation of Lamb waves in materials such as anisotropic media is complex when in a fast velocity, waves reflected from boundaries may be confounded to damage reflections [24]. Moreover, multiple modes usually exist, and their dispersive characteristics may hinder interpretation [35].

This phenomenon renders the interpretation of the received signal for defect sizing complicated. Although it is possible to detect the presence of the defect which is useful for non-destructive testing screening systems [27], mode conversion and mode mixing make any accurate quantitative prediction of the defect size impractical without resorting to advanced signal extraction and interpretation techniques [32].

In this context, ML is an advanced signal processing method that can be applied to provide defect size estimation, based on measurements related to GW phenomena [36]. As these raw physical measurements are hard to interpret [32], supervised learning paradigms can be effectively applied, given that enough input-output pairs are available for model construction and generalization. Predictive models based on ML [37] are able to make better decision making, by decreasing false alarms and providing increased computational performance [38]. In this context, it is important to note that the defect estimation, in general, requires (i) feature extraction and (ii) damage indices, which are, respectively, the input and output of the predictive model for defect estimation. Exceptions to this rule are deep learning models [39], which are able to extract the features automatically with supervised learning which demands, however, much larger datasets for the model construction phase than shallow learning. There are many works that deal with the problem of defect estimation using GW in different ways with ML, as described in a recent review [19]. In the following, we cite the most recent and relevant for the present work to establish the original contributions of the present thesis. De Fenza et al. have in [40] developed a method based on artificial neural networks for localizing and estimating defects in composite plates. In [41] the authors propose the use of a consensus ensemble modeling approach to combine different artificial neural networks models for SHM, using GW measured with piezoelectric transducers. They tested the method in detecting fatigue cracks for an aerospace application, using to that end, features constructed with baseline comparison and correlation statistics. Alguri et al. have in [42] proposed a surrogate and baseline signal strategy with dictionary learning to estimate damage in plates using GW. They employed a singular value decomposition-based k-means clustering algorithm [43] in order to generate the dictionary for sparse representation. The output of the model is the comparison of the dictionary with the baseline and its correlation-based metrics, which should be interpreted by an expert afterward. Dib et al. address in [44] the problem of SHM using a network of piezoelectric sensors with ensemble learning by consensus, where individual learners with support vector machines use different sections of the time-series measurements. The features are based on Fourier transform coefficients and experiments were carried out for estimating impact defect types under varying environmental conditions. In [45], the authors use time and frequency domain feature engineering, feature selection with evolutionary algorithms, and artificial neural network models for detecting defects in pipes. They use simulated data for training and experimental data for testing. In [46], a case study is presented for GW SHM in pipes, and preliminary results with baseline extraction are given with support vector machines. In [47] the authors employ a semi-supervised and supervised approach with principal component analysis for detecting cracks in pipes using helical ultrasonic GW. Deep learning [39] was only recently applied to SHM using GW. Its potential is great for this specific application, as it is able to automate the feature extraction process, and also deals with high dimensional datasets. In [48] the authors employed deep learning of ultrasonic waves to estimate defects on a thin metal plate, emulated by a steel washer using full wavefield measured data.

The wavefield is encoded like an image and a convolutional neural network is then used to classify the probability of defects using softmax outputs. Lamb waves were used in [17], also with deep learning architectures similarly as in [48], but using the wavelet coefficients as inputs. Finally, in [49] an active source localization is proposed, ML models predict the impact position of a steel ball in an aluminium plate using numerical simulations to create the data-driven models and experimental data to validate them.

1.2 Motivation

Based on the foregoing literature review, we can highlight some gaps that can be filled and are the motivation for this dissertation.

1. CFRP materials commonly used in the space and aviation industry [50] due to its high strength and stiffness, low weight and high fatigue resistance [51] demands extreme structural safety and reliability. As previously mentioned the propagation of waves in such materials is complex when in a fast velocity, waves reflected from boundaries in a healthy specimen may be confounded to damage reflections. Figueiredo et al. showed that system identification approaches such as Auto-Regressive (AR) models appear to be potential damage-sensitive features [52]. The usage of AR models and non-linear auto-regressive (NAR) models is not broadly used for ultrasonic GW problems. A few examples are wind turbines dirt and mud detection [53], as well as ice detection [54].

2. The most recent developments on ML applied to damage estimation with GW for corrosion-like defects demonstrated that only a few works use simulated and experimental data in order to maximize the use of information about the physical process present in high fidelity finite element simulations. This is important, as it avoids expensive measurement campaigns and provides an effective mean for creating datasets [40, 45, 49]. Moreover, none of the works described a method focused on the separation of the GW modes present in the received signals, towards more effective feature extraction.

1.3 Objectives

The main objective of this work is the evaluation of faults through the use of ML jointly with ultrasonic GW. Two case studies are explored and their specific goals are as follows.

1. To propose a methodology employed in CFRP plates that is able to detect dot damages in the latter with or without the subtraction of a baseline.
2. To propose a methodology to classify and estimate corrosion-like damages in aluminium plates. The procedure may benefit from a mode-separation technique in order to improve results.

Case study (1) is assessed through the use of experimental open-source Lamb GW data provided by [55]. The second case study (2) is accomplished thanks to the simulated and experimental data provided by the authors of [56, 57, 32].

1.4 Original Contributions

Based on case study (1), a conference paper contribution was developed. A methodology addressing dot damage detection in CFRP plates is performed. In the mentioned work, different excitation signal frequencies jointly with a subtraction of a baseline are assessed. In addition, a feature extraction procedure is applied in order to improve the ML model's performance when facing signals that do not make use of the baseline. This is a more challenging scenario closer to what is found in real-world applications.

Based on case study (2), a journal paper contribution was produced. Classification and regression perspectives regarding defects evaluation in aluminium plates are explored. In the latter contribution, a corrosion-like defect is assessed and a mode separation scheme is adopted in order to improve the feature extraction procedure and, consequently, the ML models development. It is interesting to notice that the data-driven models are trained exclusively with numerical simulations and tested with experimental data. This procedure is ideal given that experimental or real-world data is difficult to obtain and are highly costly.

Both contributions are properly listed in Appendix A.

1.5 Organization

This work is divided in four main parts.

Part I is focused on the introduction of this thesis. It is composed by Chapter 1 where a brief introduction and a literature review together with the motivation and objectives are presented.

Part II is devoted to foundations in the fields of GW and ML. Chapter 2 presents some theoretical basis for the phenomena of ultrasonic waves, themes such as SH GW and Lamb guide waves are addressed. Chapter 3 provides information on the general key aspects of ML. Subsequently, feature extraction methodologies used in the present work are discussed in Chapter 4. Chapter 5 is dedicated to the discussion of the ML data-driven models adopted. Finally, Chapter 6 is devoted to the validation methodology jointly with the metrics considered.

Afterwards, in Part III, the fundamental contributions of this thesis are given. A focus on the damage detection in CFRP plates based on system identification techniques of feature extraction is given in Chapter 7. Furthermore, in Chapter 8, the problem of classification and estimation of corrosion-like defects improved by mode-separation is tackled.

Finally, final remarks are presented in Part IV. It is composed by the single Chapter 9. It is devoted to the conclusions of the work and suggestions for upcoming activities.

Part II

Theoretical Background

2
Ultrasonic Guided-Waves

Ultrasonic GW employment is a well-stablished practice in the non-destructive evaluation/testing and SHM industries. They are primarily defined as a mechanical wave propagating through bounded structural media [19]. The fundamental equations that describe the physical phenomenon of elastic waves are the so-called wave equations. They depend primarily upon the domain where they propagate. In the fluid domain, it is described by

$$\nabla^2 p - \frac{1}{c_f^2}\frac{\partial^2 p}{\partial t^2} = 0, \quad c_f = \sqrt{\frac{K}{\rho_f}} \tag{2-1}$$

where p is the acoustic pressure that propagates in the media, K is the compressibility modulus and ρ_f is the density of the fluid and c_f is the wave speed in the fluid medium. In the isotropic solid domain, it is described by the longitudinal wave related to the scalar potential (ϕ)

$$\nabla^2 \phi - \frac{1}{c_L^2}\frac{\partial^2 \phi}{\partial t^2} = 0 \tag{2-2}$$

and to the transversal wave related to the vector potential (ψ)

$$\nabla^2 \psi - \frac{1}{c_T^2}\frac{\partial^2 \psi}{\partial t^2} = 0 \tag{2-3}$$

where the constants c_L and c_T are the speed of the longitudinal and transversal waves respectively. They are given by $c_L = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and $c_T = \sqrt{\frac{\mu}{\rho}}$ defined by the first, λ, and second, ρ, *Lamé* constants. These equations are foundations to this chapter which is dedicated to a brief theoretical base with respect to GW that propagate in plate-like objects such as (i) SH GW and (ii) Lamb GW.

2.1
SH Guided-Waves

Shear horizontal (SH) guided waves (GW) in plates have in-plane polarized vibration parallel to the surface of the plate and perpendicular to the direction of propagation [58]. Adopting the coordinate system shown in Fig. 2.1 for an isotropic homogeneous plate, SH GW propagate along the x-direction and the displacement field is polarized in the z-direction. There are, potentially, infinite SH GW modes. Each mode has a different phase and group speed and mode displacement profile through the thickness of the sample, which is how the mode's displacement field varies along the y-coordinate within the plate, i.e $-h/2 \leq y \leq h/2$, where h is the thickness of the plate. When developing the wave equations (2-2 and 2-3) considering null displacement in the x-axis and y-axis ($u_x = 0$ and $u_y = 0$), and, also zero superficial stress ($\boldsymbol{\sigma}= 0$), the displacement profile is given by:

$$u_z(y) = \begin{cases} \cos(n\pi y/h), & n = \{0, 2, 4, ...\} \qquad \text{(2-4a)} \\ \sin(n\pi y/h), & n = \{1, 3, 5, ...\} \qquad \text{(2-4b)} \end{cases}$$

where n is the mode order. Observing Eq.2-4, even-order modes have a symmetric profile across the plate thickness, whereas odd-order modes are anti-symmetric. The fundamental zero-order SH0 mode has a constant displacement along y and a phase and group speed equal to the medium's bulk shear (or transverse) wave speed, c_T for all frequencies, i.e. it is non-dispersive. All higher-order modes can propagate only for a frequency-thickness product above their respective cut-off values, given by [58]

$$(fh)_{\text{cut-off}} = nc_T/\,2 \quad , \qquad \text{(2-5)}$$

and their phase speed, c, depends on the frequency according to Eq.2-6:

$$c = c_T\, 2fh \Big/ \sqrt{4(fh)^2 - n^2c_T^2} \quad . \qquad \text{(2-6)}$$

An example of phase velocity dispersion curve for the SH modes in an aluminium alloy plate (e.g. $\lambda = 50.35\ GPa$, $\mu = 25.94\ GPa$, and $\rho = 2.71\ g/cm^3$) is shown in Fig. 8.3. It is convenient to display the dispersion curves with phase velocity versus frequency × thickness. As previously mentioned, the SH0 mode is non-dispersive, for this specific mode no matter the frequency, the phase speed remains the same.

2.2 Lamb Guided-Waves

Lamb GW combine displacements in the out-of-plane and the axial in-plane directions. Unlike the SH wave, the in-plane component is along the line of propagation [58]. In other words, they arise from a coupling between transversal and longitudinal waves reflected at the top and bottom of a plate [60]. Adopting the plate and coordinate system of Fig. 2.3, one can achieve the dispersion relations of Lamb waves [58]. Once again, starting from the wave equations (2-2 and 2-3), considering zero superficial stress ($\boldsymbol{\sigma}= 0$), and, finally, null displacement in the z-axis ($u_z = 0$), we can obtain the well-known

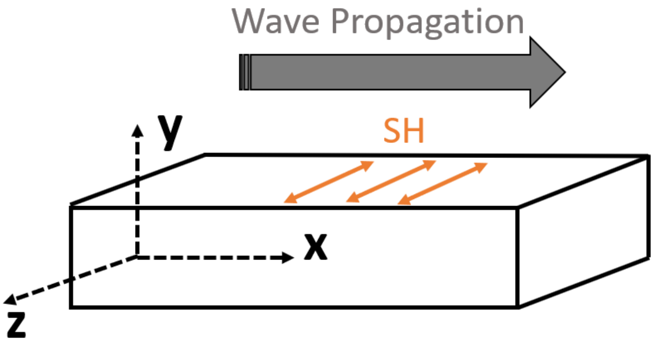


Figure 2.1: Plate representation for the SH GW. Notice how the displacement polarization field is present in the z-direction while the propagation happens in the x-axis.

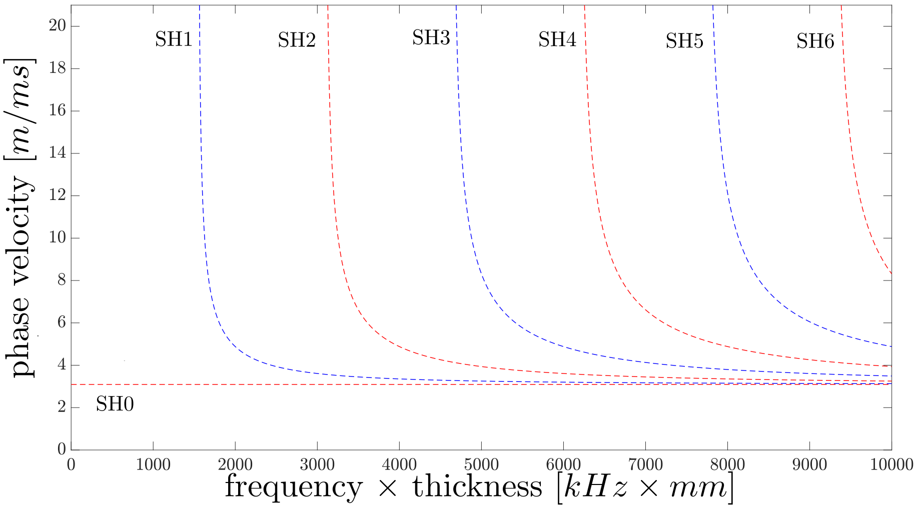


Figure 2.2: Dispersion curve of SH GW, frequency × thickness versus phase velocity, for an aluminium alloy plate. Antisymmetrical modes are represented by odd numbers while symmetrical ones are represented by even numbers, blue and red dashed lines respectively. Extracted with Dispersion Calculator software [59].

Rayleigh-Lamb equations [58]

$$\frac{\tan\left(\frac{qh}{2}\right)}{\tan\left(\frac{ph}{2}\right)} = -\frac{4\kappa^2 pq}{(q^2-\kappa^2)^2}, \qquad \text{for symmetric modes,} \tag{2-7a}$$

$$\frac{\tan\left(\frac{qh}{2}\right)}{\tan\left(\frac{ph}{2}\right)} = -\frac{(q^2-\kappa^2)^2}{4\kappa^2 pq}, \qquad \text{for antsymmetric modes,} \tag{2-7b}$$

where $p = \sqrt{\frac{\omega^2}{c_L^2} - \kappa^2}$ and $q = \sqrt{\frac{\omega^2}{c_T^2} - \kappa^2}$, ω is the frequency, κ is the wave number, c_L and c_T are the bulk velocities of the longitudinal waves and transversal waves respectively. In this case, symmetric and anti-symmetric modes are exclusively associated with in-plane displacement, regardless of being even or odd modes.

It is important to mention that the set of all ω and κ for a fixed thickness of plate (h) that solve the Rayleigh-Lamb equation results in the dispersion curve of the respective plate. As previously stated, it is convenient to display the dispersion curves with phase velocity versus frequency × thickness as we can see in Fig. 2.4. The example presents Lamb waves modes of an aluminium alloy plate (e.g. $\lambda = 50.35GPa$, $\mu = 25.94GPa$, and $\rho = 2.71g/cm^3$), notice the diverse symmetrical (S) and antisymmetrical (A) modes present in the interval up to 10000 $kHz \times mm$.

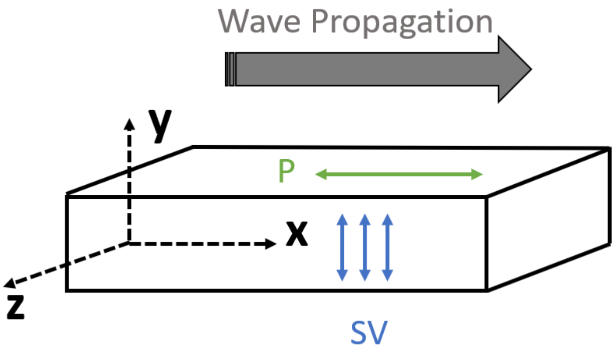


Figure 2.3: Plate representation for the Lamb GW. The combination of the displacements in the x-direction and, also, in the y-direction results in the wave propagating along the x-direction.

A major advantage of GW is the capacity of travelling in materials with high attenuation ratio and anisotropic such as carbon fiber reinforced polymers (CFRP) [24]. In this case, an anisotropic medium, the elastic constant matrix is described as [58]

$$C = \begin{pmatrix} C_{11} & C_{13} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{33} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(C_{33} - C_{13}) & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{55} \end{pmatrix}$$

(2-8)

Based on the new constant matrix, the dispersion curves of these materials have different behavior depending on the angle of propagation. Besides that, these

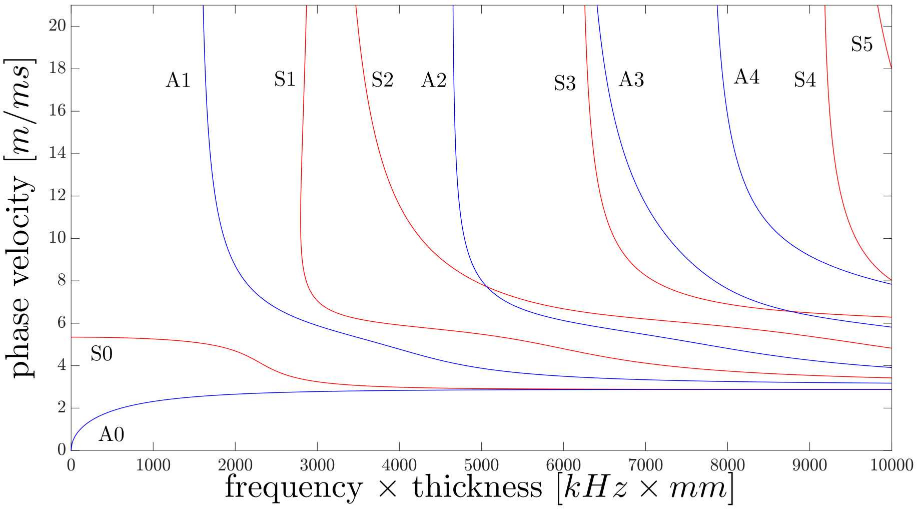
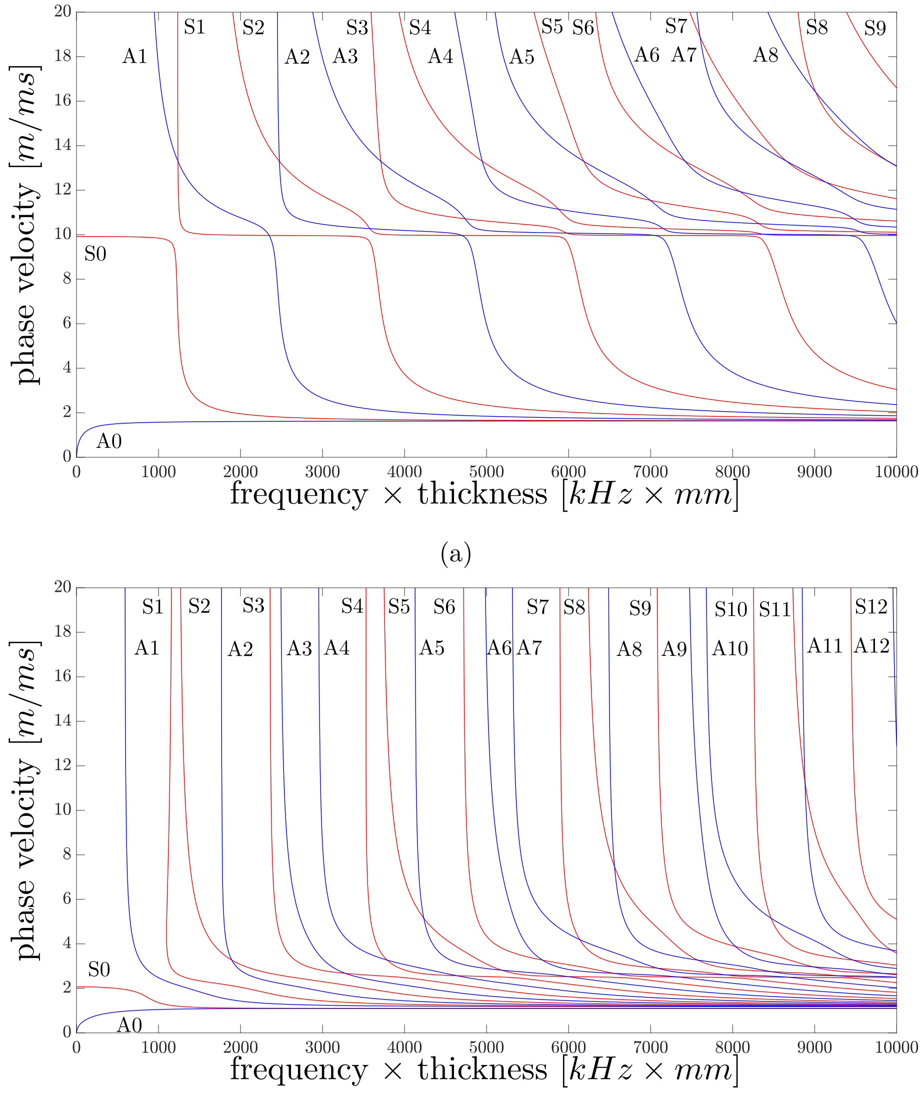


Figure 2.4: Dispersion curve of Lamb GW, frequency × thickness versus phase velocity, for an aluminium alloy plate. Antisymmetrical (A) and symmetrical (S) modes are represented by the blue and red curves respectively. Extracted with Dispersion Calculator software [59].

materials can have considerably more modes when compared to metals. An example of dispersion curve of a transversely isotropic material is seen in Fig. 2.5. The properties of the material used, a composite T800 M913 (accordingly to [59]), can be seen in Table 2.1. Notice that different dispersion curves are obtained accordingly to the propagation direction, herein represented by the 0° and 90° propagation directions, respectively Fig. 2.5(a) and Fig. 2.5(b). Additionally, one can observe a significantly greater amount of modes present in the range up to 10000 $kHz \times mm$ window when compared to an aluminium dispersion curve (see Fig. 2.4).

Table 2.1: Engineering constants of the composite material T800 M913 (accordingly to [59]).

E_1	$152.1GPa$
E_2	$6.6GPa$
G_{12}	$4.2GPa$
ν_{12}	0.252
ν_{23}	0.544
ρ	$1.55g/cm^3$



(a)

(b)

Figure 2.5: Dispersion curve of lamb waves, frequency × thickness versus phase velocity, in composite material (T800 M913 - see properties in Table 2.1. Notice the dispersion curves in the (a) 0° propagation direction and in the (b) 90° propagation direction. More than eight modes are present in the 10000 kHz × mm range. Extracted with Dispersion Calculator software [59].

3
Machine Learning

In the present chapter, general information about the machine learning (ML) paradigm is discussed. The focus is to provide enough information and knowledge to the understanding of a generic overview of the data-driven model creation process. In addition, fundamental concepts about supervised learning matter are also discussed.

3.1
Data-driven Model Creation

A view of the entire data-driven model creation is presented in this subsection. The overall process of ML model creation is given in Fig. 3.1. The pipeline of data-driven model creation normally follows the step-by-step presented, being the following:

i Data Collection: In this step, the dataset under study should be obtained. The source of the data available can be of various types, the data can be achieved by numerical means, via finite element simulations; by experimental means, via experimental setups; or either by manually or automatically sample real-world data.

ii Data Pre-processing: the second step of the pipeline is the pre-processing of data. Herein, procedures to adjust and straighten the dataset are performed. Procedures like normalizing, scaling, whitening, selection, concatenating, assembling, disassembling, and others can be used in order to better prepare the data for the fitting of models which will come further.

iii Feature Extraction: A feature extraction procedure can be adopted or not. It is useful for dimensionality reduction, if required, as well as data visualization so better insights and assumptions of the data can be made. The procedure adoption is also dependent on the models at hand. Some models (i.e. deep learning models [39]) can process raw data, in this case, the feature extraction procedure may not be necessary.

iv Target variable definition: This step is directly connected to the type of supervised learning that will be performed (classification or regression). It can be a damage index, an estimation, a binary output, and/or others.

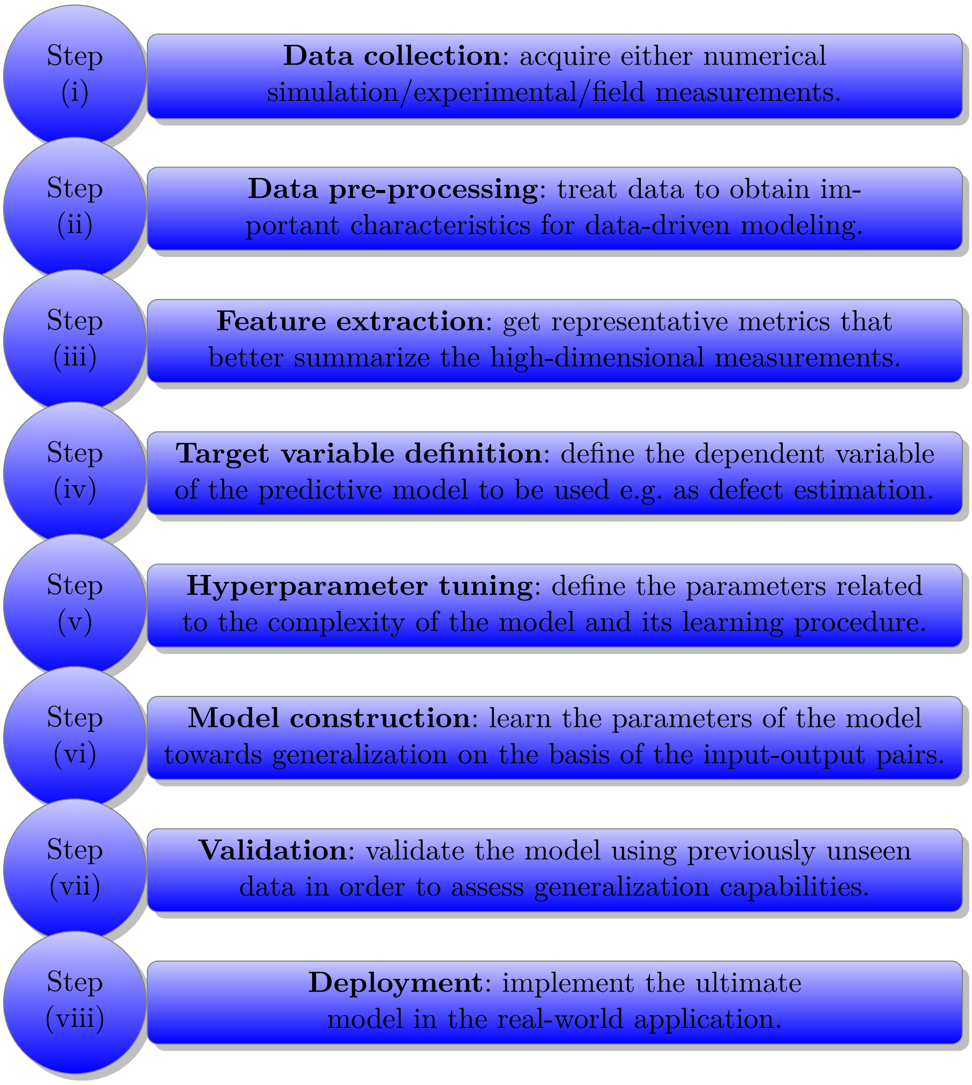
v Hyperparameter tuning: Any procedure of hyperparameter tuning should be adopted so the chances of finding an ideal model to the problem are higher. The benefits of adoption of the procedures mentioned are also improvement of the performance of the models in conjunction with avoidance of overfitting or underfitting.

vi Model creation: This step is dedicated to the construction of models capable of performing the task desired. A series of families of specific models should be considered because different data behaves differently according to the models they are fitted with.

vii Validation: In this step, the models created are tested with unseen data in some manner or adopting any cross-validation strategy. Performance metrics are then evaluated. In case of success, at the end of the current step, an ideal model is found, and the ML model maker can head to the next step. However, sometimes the expected performance is not met and the engineer is advised to retry previous steps, considering new feature extraction strategies or adopt one, retune hyperparameters or consider new data-driven models.

viii Model deployment: The final step in the data-driven model creation is the actual implementation of the model on a real-world daily basis. For this step to be accomplished, the model should be tested in real-world new data, with the same origin as the ones the model should work with regularly. It is important that every new sample can be later considered for new retraining of the model which would enable a better generalization of the model. This action is important, so the ML structure achieved is constantly improving.

At this point, a notion of the whole process of a data-driven ML model creation exists. The pipeline addressed above will be used in every contribution of this work, being the only coincidence among them the nature of the pipeline inputs or the measurements which are ultrasonic GW.



Step (i) **Data collection**: acquire either numerical simulation/experimental/field measurements.

Step (ii) **Data pre-processing**: treat data to obtain important characteristics for data-driven modeling.

Step (iii) **Feature extraction**: get representative metrics that better summarize the high-dimensional measurements.

Step (iv) **Target variable definition**: define the dependent variable of the predictive model to be used e.g. as defect estimation.

Step (v) **Hyperparameter tuning**: define the parameters related to the complexity of the model and its learning procedure.

Step (vi) **Model construction**: learn the parameters of the model towards generalization on the basis of the input-output pairs.

Step (vii) **Validation**: validate the model using previously unseen data in order to assess generalization capabilities.

Step (viii) **Deployment**: implement the ultimate model in the real-world application.

Figure 3.1: Predictive modeling general procedure. The model can be reiterated according to model validation metrics going back to any of the previous steps. Note that the step-by-step implies thus recurrence as the data-driven modeling is inherently iterative.

3.2 Regression and Classification Problems

There are many definitions of supervised learning. Perhaps, the simplest one, or, the most straight forward was given by Haykin [4], he defines supervised learning as learning with a teacher. In other words, it is the process of creating a model which estimates a condition of an unknown sample based on known input-output pairs of training data [2].

The supervised learning process can be depicted in two main approaches based on the model output: regression or classification. The former can be characterized when the output is continuous or gradual. An amplitude can be portrayed by this approach. Fig. 3.2 exhibits an example of a function which maps a regression problem. One can expect that the continuous predicted values approximate the actual known values.

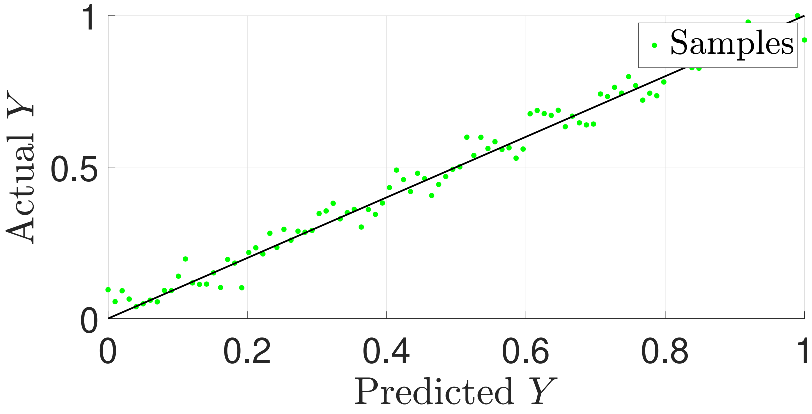


Figure 3.2: Simplified regression problem. The target of the model is given so that the predicted value (x-axis) is as closest as possible to the actual value (y-axis), the actual-predicted pairs are represented by the green dots. Notice that the diagonal line represents an ideal regression model, every sample which falls in this particular line has a perfect relation between predicted and actual outputs.

The classification problem can be defined when a discrete target is delivered, classes or categories can be tackled by this strategy. An example of the classification concept is depicted in Fig. 3.3 where a simple classification problem is treated with only two attributes and only two classes (Attribute 1 or Attribute 2, e.g. defect or nominal). Notice that this concept can be extended for a more complex problem where a line would not be able to accomplish the task. Therefore, for instance, with more attributes, the line evolves to a hyperplane. Additionally, the problem may embrace more classes or categories, namely a multi-class problem.

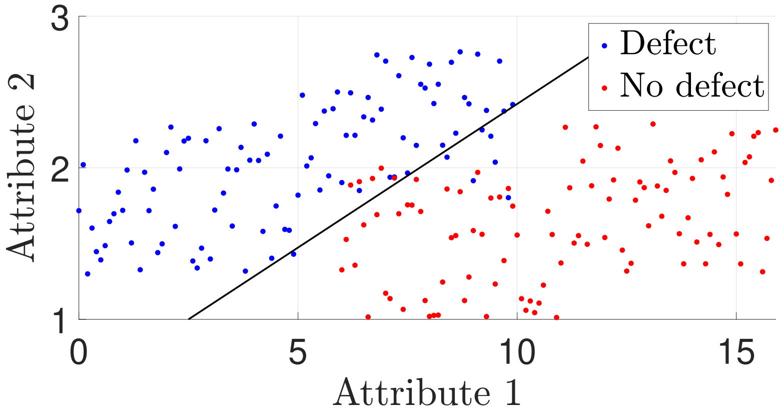


Figure 3.3: Simplified classification problem. Notice that the problem stated maps a function (black line) which divides a group of samples in two classes: defect (blue dots) and no defect (red dots).

4
Feature Extraction

This chapter is dedicated to defining all of the methods of feature extraction adopted in this dissertation. Feature extraction is one of the main steps in the ML model development pipeline. The procedure is a technique that transforms and simplifies data [61], its use normally grants to the data-driven model development benefits that are key to this study: (i) dimensionality reduction, the raw data, or the data as it is provided, has dimensions that are either not manageable when provided to the training of the models or are either too computational expensive since many attributes should be accounted; (ii) data understanding and visualization, sometimes the feature extraction can deliver important information or insights from the data at hand after the procedure is performed. Several are the feature extraction approaches such as extraction of local features [62], convolutional methods using kernel and syntactic and structural methods [63], as well as linear and non-linear space embedding methods [64], techniques to project or embed the data into lower-dimensional spaces while conserving the maximum original information [65], discrete Fourier transform [66], wavelet transforms [67, 68], and many others. In this work, two types of feature extractions were used, feature extraction based on system identification techniques [69], and temporal and statistical features [70].

4.1
System Identification Techniques

System identification is a method of mathematically describing a system by processing the observed inputs and outputs of the system [71]. Some methods based on the least-squares algorithm are sometimes used in the feature extraction task [52, 53, 54]. In this work, we adopt the linear and non-linear Auto-Regressive Models.

4.1.1
Auto-Regressive Models

Consider a discrete signal $y(k)$ as follows [69]:

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) - ... - a_{n_a} y(t-n_a) + \xi(k) \tag{4-1}$$

where $\xi(k)$ represents measurement error, offset or noise; n_a is a non-negative integer indicating the order of the model; and a_i are the coefficients of the AR polynomial.

Assuming that the model is linear, one can use the batch least squares algorithm. Being the regression vector $\phi(k)$, whose dimension is n_a, defined as:

$$\phi(k) = \begin{bmatrix} -y(k-1) & -y(k-2) & \dots & -y(k-n_a) \end{bmatrix}^T \quad . \tag{4-2}$$

The model parameters are then defined as:

$$\boldsymbol{\theta} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n_a} \end{bmatrix}^T \quad . \tag{4-3}$$

Equation (4-1) can be rewritten as a linear regression model [69]:

$$y(k) = \phi^T(k)\boldsymbol{\theta} + \xi(k) \quad , \tag{4-4}$$

and the AR model matrix form can be obtained with

$$\mathbf{y} = \boldsymbol{\Phi}_{\mathbf{AR}}\boldsymbol{\theta} + \boldsymbol{\xi} \quad , \tag{4-5}$$

where $\mathbf{y} = \begin{bmatrix} y(p) & y(p+1) & \dots & y(N) \end{bmatrix}^T$, is the output measurement vector with $p = n_a + 1$ is an auxiliary variable for the maximum order of the model and N is the length of measurements are made, $\boldsymbol{\Phi}_{\mathbf{AR}}$ is the regression matrix of dimension $N - p \times n_a$. In this way, the model can be conveniently estimated using any standard least squares solver.

4.1.2 Non-Linear Auto-Regressive Models

The NAR models can be defined as follows [71]:

$$y(k) = F[y(k-1), y(k-2), ..., y(k-n_a)] + \xi(k), \tag{4-6}$$

where $F[.]$ is a nonlinear function. A commonly used nonlinear function is adopted in this work, the power-form polynomial representation:

$$\begin{aligned} y(k) = \theta_0 + \sum_{i_i=1}^{n} \theta_{i_1} y(k-i_1) + \sum_{i_1=1}^{n}\sum_{i_2=i_1}^{n} \theta_{i_1 i_2} y(k-i_1)y(k-i_2) + \\ ... + \sum_{i_1=1}^{n} ... \sum_{i_l=i_{l-1}}^{n} \theta_{i_1 i_2 ... i_l} y(k-i_1)y(k-i_2)...y(k-i_l) + \xi(k), \end{aligned} \tag{4-7}$$

where l is the degree of polynomial nonlinearity and $\theta_{i_1 i_2 ... i_m}$ are model parameters.

The NAR model matrix form can be summarized as

$$\mathbf{y} = \boldsymbol{\Phi}_{\mathbf{NAR}}\boldsymbol{\theta} + \boldsymbol{\xi} \quad . \tag{4-8}$$

The total number of terms in the full NAR models (M) is defined as following

$$M = \frac{(n_a + l)!}{n_a! l!} \quad . \tag{4-9}$$

Note that the quantity of terms in the model is related to the orders l and n_a.

In both cases, linear and non-linear approaches, the estimated parameters of the model ($\boldsymbol{\theta}$) is utilized as features of the respective samples.

4.2 Temporal and Statistical Features

Temporal and statistical features for signals in the time domain are also adopted in this work. This approach is advantageous as it is simpler and less computationally intensive. They have been used elsewhere, mainly in health monitoring applications such as [20, 70], and carry basic time history information, as well as basic statistics. Based on an n-sample discrete signal $x(i)$, examples of time features are L_2-energy norm ($||x||_2$), namely, the square root of the total energy of a signal, defined as [70]

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x(i)^2}, \tag{4-10}$$

and; also, the curve length (L) [72]

$$L = \sum_{i=1}^{n} |x(i) - x(i-1)|. \tag{4-11}$$

Statistical features such as Kurtosis (Ko), which describes the shape or peakness of a distribution [73]

$$Ko = \frac{\frac{1}{n}\sum_{i=1}^{n}\left[x(i) - \bar{x(i)}\right]^4}{\left\{\frac{1}{n}\sum_{i=1}^{n}\left[x(i) - \bar{x(i)}\right]^2\right\}^2} \tag{4-12}$$

and; also, the Pearson Coefficient which is the correlation coefficient between two different signals, such as the incident wave (x_i) and the reflected wave (x_r)

$$\frac{\sum_{i=1}^{n}\left[x_i(i) - \bar{x_i(i)}\right]\left[x_r(i) - \bar{x_r(i)}\right]}{\sqrt{\sum_{i=1}^{n}\left[x_i(i) - \bar{x_i(i)}\right]^2}\sqrt{\sum_{i=1}^{n}\left[x_r(i) - \bar{x_r(i)}\right]^2}} \tag{4-13}$$

are also explored.

5 Data-driven Models Architectures

In this chapter, we discuss ML data-driven models that are considered in this work. There is a considerable number of models available when discussing the supervised learning task. In the literature, this decision is usually made based in a group of models of different families. In [74], a variety of classifiers are evaluated facing various datasets in order to ultimately finding the best model. In this work we consider two types of ML models. (i) Linear Models and (ii) Artificial Neural Networks - Multilayer Perceptrons.

5.1 Linear Models

Linear Models (LM) are the simplest predictive models to build. They are normally used to allow comparison and assess the difficulty to solve the problem [75].

Considering x_j as an input of the j-th feature, being the total number of features equal to m and $\hat{y}$ the predicted output of the model, the simple linear model for regression is given by

$$\hat{y} = w_0 + \sum_{j=1}^{m} w_j x_j. \tag{5-1}$$

where $\boldsymbol{w} \in \mathbb{R}^{m+1}$ are the coefficiets of the model that should be learned from the data. This model can be trained with the Penrose-Moore pseudo inverse, which amounts to the least squares. Note that for evaluating the classification of the output of the linear model for multiple classes, it is advantageous to consider linear discriminant analysis or logistic regression [75]. Logistic regression is basically an adaptation of the linear regression. It is advantageous as it can output probabilities. It is mathematically defined as [76]

$$ln\left[\frac{p}{1-p}\right] = w_0 + \sum_{j=1}^{m} w_j x_j. \tag{5-2}$$

being the left side function defined as *logit* where p is the probability of a given input belonging to class $y = 1$. We can solve the logit equation to p obtaining

$$p = \frac{e^{w_0 + \sum_{j=1}^{m} w_j x_j}}{1 + e^{w_0 + \sum_{j=1}^{m} w_j x_j}}. \tag{5-3}$$

Notice that the value of p can be defined as a number between 0 and 1. Therefore, when $p > 0.5$ the $\hat{y}$ is defined as class 1 ($\hat{y} = 1$). Otherwise, $\hat{y}$ is defined as class 0 ($\hat{y} = 0$). One can see that the output considers a binary classification problem. In order to consider a multi-class situation, the logistic regression can be generalized using a one-versus-rest (OVR) approach. Values of $\boldsymbol{w}$ can be defined minimizing the logistic regression loss function $J(\boldsymbol{w})$ defined as

$$J(\boldsymbol{w}) = -[p(\boldsymbol{x}) \times y + (1 - p(\boldsymbol{x})) \times (y - 1)]. \tag{5-4}$$

Methods like the gradient descendent [77] or newton raphson [78] can be used in order to solve the optimization problem above-mentioned.

5.2 Artificial Neural Networks

An ANN is a mathematical model based on the functioning of biological neural systems. Simply put, it is a group of layers of neurons that are bonded and inter-connected, an adaptive system which changes its structure based on information that flows through the network during a learning process [79].

There are many types and variants of ANNs, from more complex ones such as convolutional neural networks and recurrent neural networks [39] to ordinary models like feedforward neural networks, more specifically multilayer perceptrons neural networks (MLP) which will be the focus of this subsection.

The workings of an MLP is simple. It is composed of neurons, its simplest piece. A neuron can mathematically be represented by the following equation [4]

$$y_k = \phi\left[b_k + \sum_j w_{kj} x_j\right] \tag{5-5}$$

where $x_1, x_2, ..., x_n$ are the input signal; b_k is an external parameter called bias; $w_{k1}, w_{k2}, ..., w_{kn}$ are the synaptic weights of the neuron k; y_k is the output; and $\phi(.)$ is the activation function. The latter being responsible for the nonlinearity of the model, which can be, for example, a threshold function, a sigmoid function, a hyperbolic tangent ($tanh$), a Rectified Linear Unit ($ReLU$), and many others. A group of neurons orderly disposed is named a layer, a MLP is by definition composed by two or numerous layers. The greater the number of neurons and layers, the more complex an ANN is. A mathematical representation of a network with many layers is synthetized by

$$\hat{y}_j = \phi\left[\sum_r w_{sr} \phi\left[\sum_l w_{pl} \phi\left[\dots \phi\left[\sum_j w_{kj} x_j\right]\right]\right]\right] \tag{5-6}$$

where s, p and k represent neurons from different layers associated with different node connections from different layers given by r, l and j, the bias term which was previously mentioned in Eq. 5-5 was ignored here for simplification.

All the weights and bias of the MLP are optimized as far as the network is trained, in a manner that by the end of the learning process the model is capable of correctly estimating certain output given any input. In order to accomplish that, a training process is necessary. The most common training procedure is based on the backpropagating of errors [80], a procedure which in each, namely, epoch of training, the error, or the difference between the predicted output and the actual values, is considered. The error backpropagates correcting the values of the connected weights and bias, this step is repeated in every epoch, so the error is constantly being minimized until a stopping criterion is met. Many optimizers are used in the training of these algorithms. Common optimizers are the classical, stochastic gradient descent (SGD); and, more recent and robust ones such as the Adaptive Moment estimation (Adam) [81]. MLPs can work in many types of problems, the differences are mainly in the loss function considered and, also, in the activation function of the output layer. For instance, a classification problem could work with a softmax activation function [82] in the output layer, mathematically defined as

$$\hat{P}_n(x) = \frac{e^{s_n(x)}}{\sum_{j=1}^{N} e^{s_j(x)}} \tag{5-7}$$

where N is the number of classes, $s(x)$ the value of the output of the model for sample $\mathbf{x}$, and $\hat{P}_n(\mathbf{x})$ is the predicted probability of the sample $\mathbf{x}$ belonging to class n. This probability is presented to a categorical cross-entropy (CE) loss function [83], given by

$$CE = -\sum_{j=1}^{N} y_j(x) \log\left(\hat{P}_j(x)\right) \tag{5-8}$$

where y_j is the real output value for class j (given by 0 if it is not class j and 1 if it is), and $\hat{P}_j(\mathbf{x})$ the predicted probability of sample $\mathbf{x}$ belonging to class j. A regression problem, however, would work with a linear activation function in the output layer jointly with a mean squared error as a loss function. A simple example of MLP is depicted in Fig. 5.1. One can notice that it is a network for an input with three attributes since it has the same number of neurons in its input layer. It is also composed by a single hidden layer with a *ReLU* activation function. The model evaluated a binary problem where only two types of classes can be estimated given the number of neurons in the last or output layer.

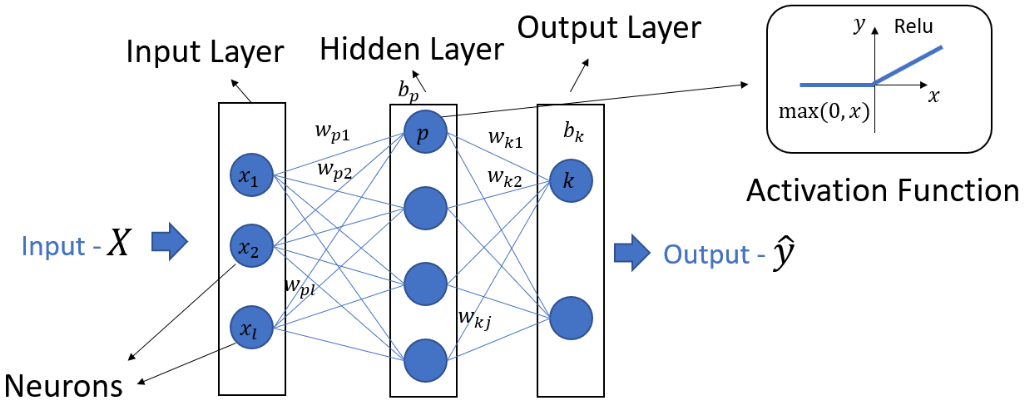


Figure 5.1: A simple MLP. Notice that the emphasized example has a basic architecture. One hidden layer with a *ReLU* activation function attached to an output layer which can estimate two types of classes.

6
Validation

In order to evaluate the performance of each of the data-driven trained models, a reliable validation strategy should be adopted. Ideally, the model is validated with data that has not been used in the learning of the process, after all, we are interested in the model's capability of generalizing, in other words, to provide an unbiased sense of model effectiveness [37]. The validation process can be focused on (i) the model validation strategy, (ii) the model validation metrics, or, yet the (iii) hyperparameters tuning procedure which are inherently associated with the validation strategy herein adopted. The current chapter is dedicated to all the aforementioned topics.

6.1
Resampling-based Model Validation

The most common technique of ML model validation is the holdout method. This approach consists of randomly divide the available dataset into two subsets, called the training dataset and the testing or validation dataset. As the name suggests, the first one will be responsible for fine-tuning the model and the latter to validate. Hence, the test data will be used to check the performance of the already trained model, the metrics of performance met are then used to make the inferences about the model.

This strategy is a good option when signals from different natures are available, for example, simulated and experimental signals. Part of the dataset can then be used solely for the construction of the models (simulated signals) and part of it is adopted to the validation (experimental signals).

However, the latter may not be the most adequate for samples of the same nature. The validation process is subject to a single split which may result in erroneous or false estimation of the model, due to the sampling of testing outliers for example. Besides that, the test dataset may have important information which is not accounted for in the fitting procedure [84].

In order to accomplish an effective validation using all the data available and drawing statistically significant results, the cross-validation (CV) strategy may be adopted. The CV or, also called, k-fold cross-validation, is the validation procedure which basically considers the holdout method more than one time.

Instead of splitting the dataset available once, composing two groups of test and train samples, the dataset is divided into k-folds. The procedure is summarized in Fig. 6.1, the division of the dataset made considering the empirical k adopted is shown. Normally, each fold has approximately an equal number of samples, the model is fit k times. In each fit, the training set is composed of $k-1$ folds and the test is held with the remained fold.

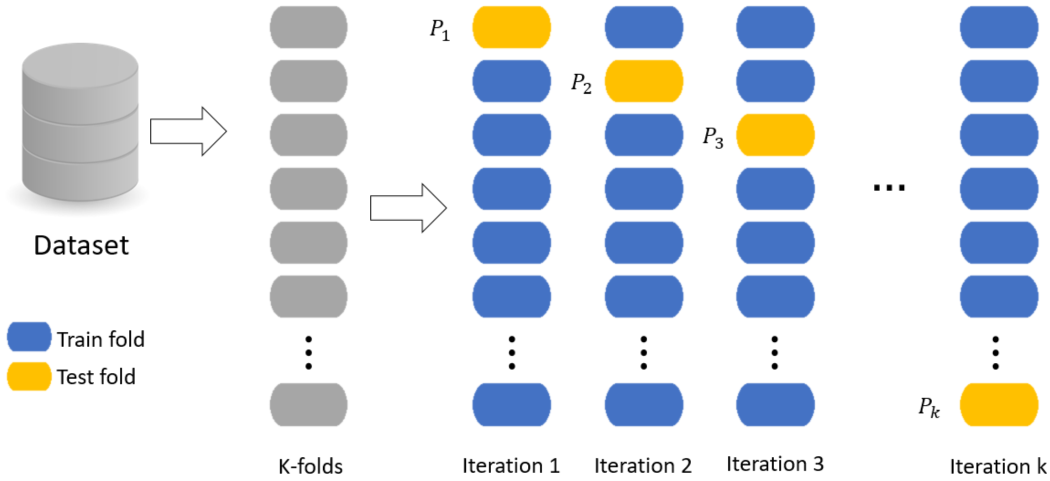


Figure 6.1: Division of dataset for k-fold CV procedure. In each of the k iterations the $k-1$ folds are used in the training process while the remainder folder is used to evaluate the performance of the model.

In the end, the performance of the model is then evaluated based on the average of each of the performance metric results of the k tests performed. It is interesting to mention that there are many schemes in which the fold division can be created, such as stratified k-fold, where each fold preserves the proportions of classes of the original dataset; leave-one-out (LOOCV), especially used in smaller datasets, where the dataset is divided in N folds where N is the number of existent samples; and, also, repeated k-fold, where the k-fold procedure is repeated in order to consider randomness that models with the same architectures and hyperparameters may have.

6.2 Hyperparameter Tuning

ML models are structures which are set or built according to a training process. Many of these parameters are tuned during this process, however, some of them are previously established in the implementation of the algorithm, they are designed, most of the time, empirically, thus being directly related to the nature of the data in question. The latter parameters are also called hyperparameters, they are terms that are not trainable so a strategy to define them is needed.

We can divide the hyperparameter setting techniques into two main types: (i) surrogate models and (ii) exhaustive searching. Concerning surrogate models, one can use optimization techniques such as gaussian processes as well as bayesian algorithms to tune hyper-parameters [85]. However, the evaluation of more than one model with different types of hyperparameters (categorical, continuous, etc) makes the approach impractical [86].

Concerning the exhaustive searching techniques, two main approaches are known and depicted in Fig. 6.2. They are (a) Grid Search which evaluates the model under discussion in all the possible combinations within a universe of hyperparameters, and, also, (b) random search which performs the training of the model only in a specific number of random combinations. One can use the traditional grid search which is basically an extensive search through a whole universe of combinations provided by the ML model maker.

Another approach of exhaustive technique is the random search. This technique evaluates a specific number of random combinations within a given universe of hyperparameters. Both approaches are useful, however, the random search has shown to be a more efficient option empirically and theoretically as demonstrated by [87].

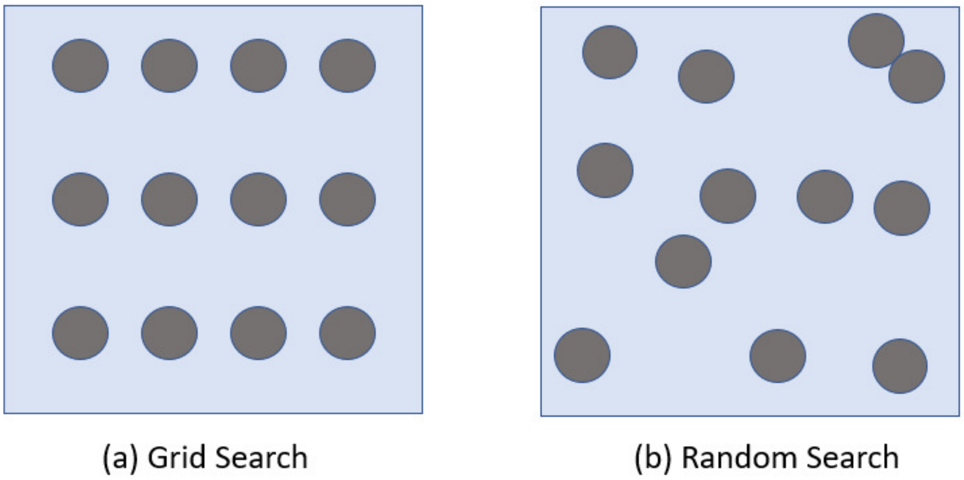


Figure 6.2: Exhaustive searching techniques [86]. Notice that the boxes represent universes of hyperparameters where (a) Grid Search evaluates the model under discussion in all the possible combinations within the universe while (b) random search performs the training of the model only in a specific number of random combinations.

6.3 Model Validation Metrics

In this subsection, the evaluation metrics adopted in the analysis of the models discussed in this study are presented. Validation metrics are key indicators that provide a numerical sense of the performance of the model to the decision-maker. Additionally, it also enables the ranking and comparison of different hyperparameter combinations and algorithms.

In order to compare the results of both perspectives, classification, and regression, different metrics are adopted. These will be better explained in the following subsections.

6.3.1 Classification Metrics

In this work, we focused on two main metrics for the classification analysis. The first one is the accuracy which is the rate at which the model correctly classifies a group of test samples. A general mathematical representation is given by [88]

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{6-1}$$

where TP, FP, TN and FN stand for True Positive, False Positive, True Negative and False Negative respectively in a classification problem.

A second metric considered in the classification tasks is the kappa coefficient (Kc). It is used to analyze the inter-rater agreement of the model [89]. In other words, it tells one if the model is in fact correctly classifying the data as intended or if it is correctly classifying by chance. The closer to 1 the value of this coefficient, the more reliable the model is. It is mathematically defined by the probability of observed agreement (P_o) and the probability of agreement by chance (P_c) [90]

$$Kc = \frac{P_o - P_c}{1 - P_c}, \tag{6-2}$$

for ML models performance the P_o is given as the ACC while the P_c is mathematically defined as

$$P_c = \frac{(TP + FP) \times (TP + FN) + (TN + FP) \times (TN + FN)}{(TP + FP + TN + FN)^2}. \tag{6-3}$$

6.3.2 Regression Metrics

Concerning the regression problem, different metrics must be assumed since in this approach the models do not classify the samples, i.e. outputs are not discrete numbers or classes. In this context, models output continuous values or magnitudes.

The first metric, the $RMSE$, is, by definition, the square root of the mean of the square of the difference between the actual and predicted values of a discussed model. It tells how much the set of predicted values are further to the actual values. This metric is mathematically defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6-4}$$

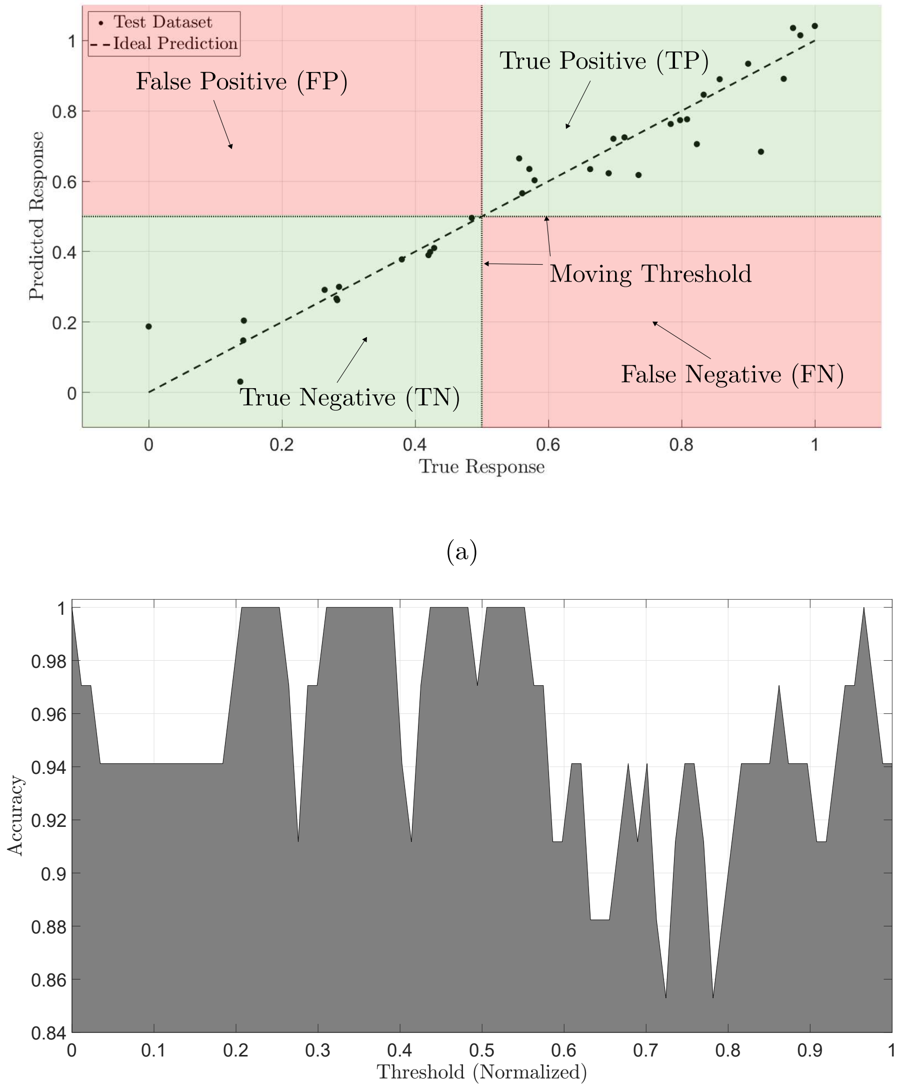
where y is the real output value while the $\hat{y}$ is the predicted output. In summary, the closer to 0 the $RMSE$, the better it is.

Another metric adopted in this work is the area under the accuracy curve ($AUAC$). It is especially useful in order to compare the results of the regression and classification formulations, it is possible to measure the accuracy of a binary classifier by setting a moving threshold for determining nominal and faulty conditions. This is important, as the regression does not need to be exact. In other words, it provides more information as a continuous output, but in many cases, only a binary decision has to be made. Fig. 6.3(a) illustrates the concept. It is possible to see that the accuracy metric varies with respect to the moving threshold. The AUAC can be calculated as is seen in 6.3(b), note that the area in grey is the AUAC. The closer the AUAC is to one, the better.

Lastly, one more metric adopted for the regression model is the R^2. It measures the correlation of the output values of a model and the actual values. The closer to 1.0 the R^2 value, the better is the performance of the model. The metric is denoted by

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{6-5}$$

where $\bar{y}$ is the mean of actual observations.



(a)

(b)

Figure 6.3: AUAC example. (a) Regression curve of true versus predicted response. The regression can be evaluated as a binary classification. According to a moving threshold, accuracy can be calculated. For instance, a moving threshold of 0.5 would result in perfect accuracy. (b) Accuracy as a function of threshold (normalized). The AUAC is the area in grey which assists as a metric of validation. The closer the AUAC is to one, the better it is.

Part III

Contributions

7
Damage Detection in Composite Plates with Ultrasonic Guided-waves and Nonlinear System Identification

In this case study, a SHM strategy to detect flaws in CFRP plates using ultrasonic GW signals was employed. Signals are processed for feature extraction based on linear and nonlinear auto-regressive system identification techniques and artificial neural networks. Additionally, the subtraction of baseline is considered and results have shown that perfect detection was able to be achieved in the former case. This shows that the present contribution is suitable to be used for SHM applications of CFRP plates subject to damage and encourages further research work to improve the baseline-free case.

7.1
Problem Description

The dataset utilized in this case study is open and accessible at `http://openguidedwaves.de/` [55]. Experiments were performed in a plate of CFRP with the dimensions 500 mm x 500 mm and 2 mm thickness. A climate chamber was used in order to regulate the experimental environmental conditions such as the temperature and relative humidity (RH) which were both set to 23°C and 50% respectively. It is essential to use a climate chamber because the propagation speed of ultrasonic waves is affected by the environment temperature and humidity [91, 92]. The specimen was equipped with 12 piezoelectric transducers in a pitch-catch configuration where any transducer could be employed as a transmitter while the others were acting as receivers. Each transmitting transducer was excited with a 5 cycle Hann-windowed tone-burst with 100 mV and frequency ranging from 40 kHz to 260 kHz. Due to this excitation, ultrasonic Lamb GW were generated and propagated in the CFRP plate [58]. The sampling frequency was equivalent to 10 MHz, the time-window available is 1300 μs.

A series of experiments were performed in order to obtain 60 nominal measurements, that is to say, obtained in a non-defective plate, and 30 defect measurements. Each defect is employed reversibly, in other words, the presence of the defect in the plate was emulated by placing an aluminum disk on the plate's surface, bonded with an adhesive, as proposed by the authors in [93].

Damages are labeled according to the position of the aluminum disk, as shown in Fig. 7.1. There are 28 single defects, for instance, damage D_1 which is related to the aluminum piece solely in position 1. There are two multi-defect cases, damage D_{25}/D_{28} and $D_{14}/D_{25}/D_{28}$. These are represented by disks simultaneously placed in positions 25 and 28, and 14, 25, and 28 respectively.

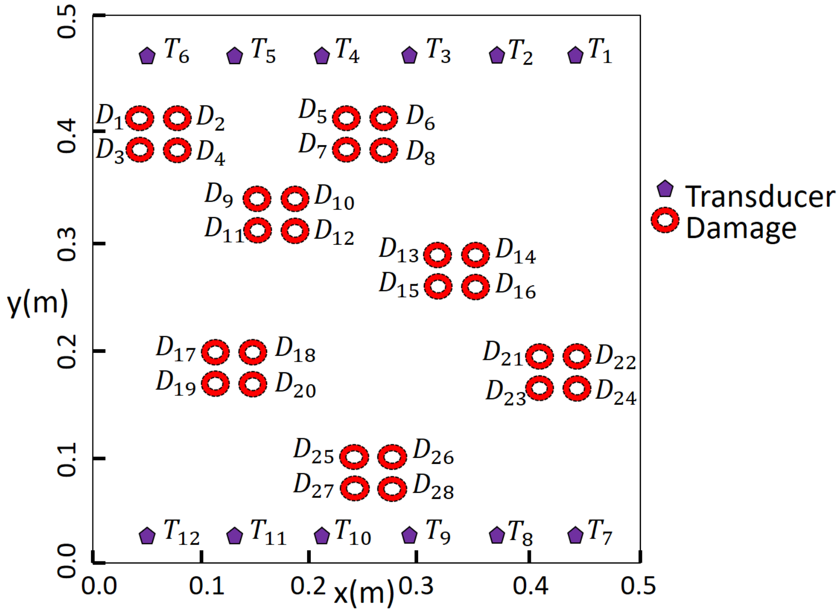


Figure 7.1: Schematic of the CFRP specimen plate, based on [55]. The transducer positions T_1 to T_{12} are represented by the purple pentagons and defect locations D_1 to D_{28} are represented by the red discs.

All signals were filtered using a third-order high-pass Butterworth filter with a cut-off frequency of 20 kHz. The network is composed of all the possibilities of channels among the 12 transducers available, i.e., 66 possibilities in total. Further information or description of the database and experiment are available in reference [55].

7.2 Method

A network of transducers composed of 36 channels is adopted. The network amounts to all possible pairwise combinations of transmitters ($T_1 - T_6$) and receivers ($T_7 - T_{12}$). Namely, we used the signals from the following pairs: $T_1 - T_7$, $T_1 - T_8$, ..., $T_1 - T_{12}$, $T_2 - T_7$, $T_2 - T_8$, ..., $T_6 - T_{11}$, and $T_6 - T_{12}$. The remaining channels are ignored as they may present redundant information. The reader is encouraged to observe a nominal (obtained in a non-defective plate) sample signal acquired in all of the aforementioned channels in Fig. 7.2. One can notice that different amplitudes and shapes due to the dispersion and end-reflection are depicted for each of the diverse paths.

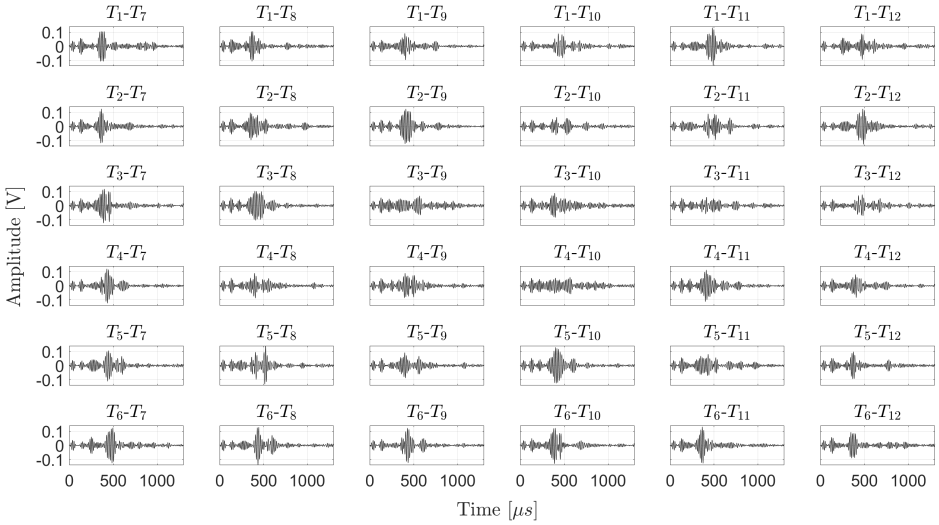


Figure 7.2: Signals received due to 60 kHz excitation in a nominal-case plate for several transmitter-receiver pairs, labeled at the top of each plot. Refer to Fig. 7.1 to the positioning of the transmitter and receiver transducers. Notice the different amplitudes and shapes for the several signals due to dispersion and end-reflection at the different propagating paths.

In this work, three excitation frequencies were arbitrarily elected, namely, 60 kHz, 140 kHz, and 240 kHz. Signals were presented to four modules which synthesize the methodology adopted herein, described in Fig. 7.3.

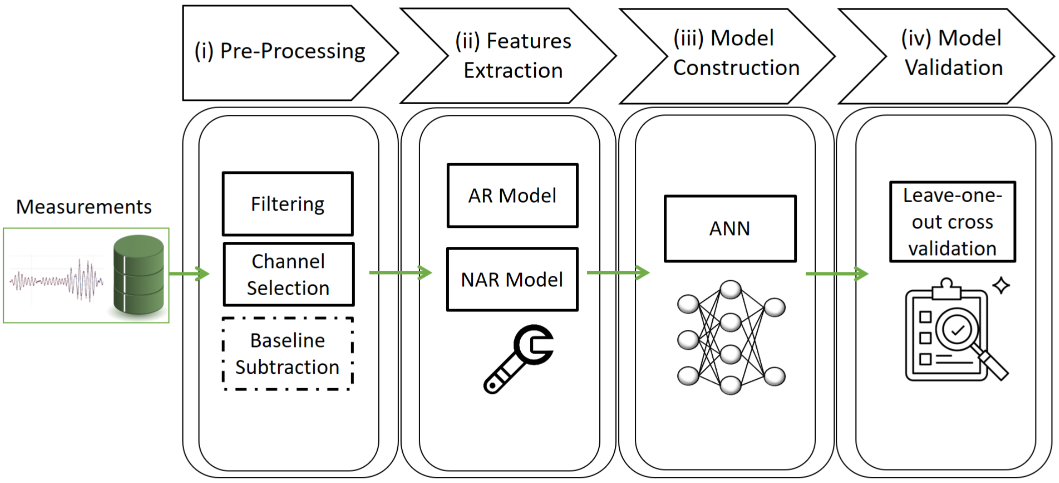


Figure 7.3: Overview of the adopted methodology. The feature extraction is composed of either AR or NAR models. It is important to mention that the measurements can make use of the baseline or not in the devised process, which is referred to by a dashed box.

Module (i) is composed of data pre-processing. In this stage, the transducer's network of interest is defined, the channels considered are selected and filtered. Afterward, the 60 nominal data are randomly divided into two groups.

The first one, composed of 30 nominal cases, is used to define a nominal mean which will be called baseline and will be further used to facilitate the distinction of the defect and nominal situations. The second group, also composed of 30 nominal cases, is part of the database for feature extraction. A balanced dataset is then composed of 30 nominal samples and 30 defect samples.

Another procedure of the pre-processing module is the baseline subtraction (represented by the dot-dash line in Fig. 7.3), an optional operation that facilitates distinguishing the signals captured by receivers. Fig. 7.4 shows an example of the baseline subtraction. The blue line in Fig. 7.4(a) represents the signal $D_{14}/D_{25}/D_{28}$ with no nominal mean subtraction (baseline subtraction). Although it is a signal of a flawed situation, it is very similar to the nominal case (red dashed line) and the baseline (black dotted line). However, imposing the baseline subtraction, the damage signal $D_{14}/D_{25}/D_{28}$ (purple line in Fig. 7.4(a)) reveals that the wave packet from 400μs to 500μs is the most affected by the presence of the defect, being, thus, clearly identified after subtraction, but is hardly noticeable before. The subtraction of a nominal case from the baseline (green dashed line in Fig. 7.4(b)) results in a signal with mostly noise. In other words, the baseline subtraction shows to be an adequate strategy to distinguish nominal and damage signals.

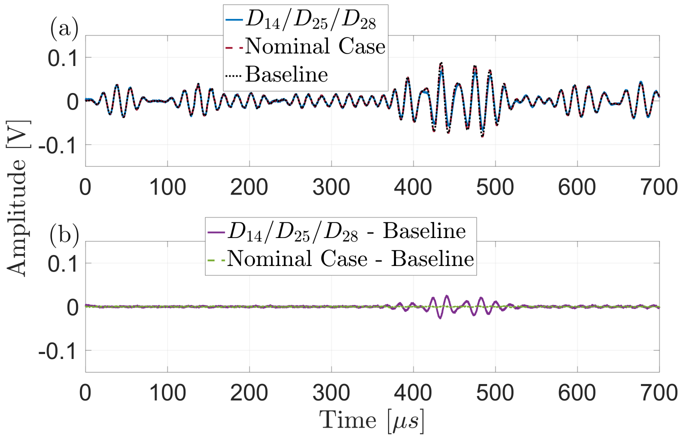


Figure 7.4: Example of 60 kHz signal - Transducer pair: 1 - 10 (T_1-T_{10}). (a) Damage $D_{14}/D_{25}/D_{28}$ and Nominal case with no Baseline subtraction. Baseline signal in black. (b) Damage $D_{14}/D_{25}/D_{28}$ and nominal case with baseline subtraction. The baseline subtraction facilitates the differentiation of a damage signal.

Six datasets were constructed, two for the 60 kHz frequency, two for the 140 kHz frequency, and two for the 240 kHz, each of them is divided accordingly to the optional subtraction of the baseline and no subtraction of baseline. Hence, datasets are defined as: *60 kHz raw*, *60 kHz baseline subtracted*, *140 kHz raw*, *140 kHz baseline subtracted*, *240 kHz raw*, and *240 kHz baseline* *subtracted.*

Module (ii) is the features extraction module. In this paper, AR and NAR models are used as the feature extraction method, details of the implementation of the latter can be found in subsections 4.1.1 and 4.1.2. Module (iii) builds the classification model based on the training of the data utilizing an ANN which can be referred to in section 5.2.

Finally, in module (iv), the model is validated. The approach adopted is the Leave-One-Out Cross-Validation (LOOCV) [94]. It consists of training all the data but one removed sample which will be used for testing. As the dataset is relatively small, this validation procedure is more adequate than using a percentage of holdout data.

7.3 Results

The performance analysis discussed in this section addresses the results achieved on the Composite Plate problem case study (Section 7.1). In summary, it deals with the choice of hyperparameter n_a of the AR model, as well as the classification rate obtained by the trained models. The consideration of the baseline is assessed, likewise, different ANN architectures. In order to find an optimum architecture of ANN for each dataset available, the number of hidden layers and neurons were varied. The hidden layers varies in quantity, either 1 ({X}), 2({X X}), or 3({X X X}). In addition, the number of neurons composing these layers were also diversified, hidden layers with 10 (X=10) and 100 (X=100) neurons were considered. The activation function of the output layer for the model is a softmax [82]. The scaled conjugate gradient backpropagation (SCG) learning algorithm is adopted [95]. All the supervised learning models were executed 30 times with different random seeds. The AR and full NAR models have been estimated using the Penrose-Moore pseudo inverse.

7.3.1 Hyperparameter Choice

Initially, the AR parameter n_a was defined empirically based on the iterations in the range between 1 and 15. The validation metric used was the RMSE. Since we are assessing the one step ahead approach [69], the greater the n_a, the lower the values of RMSE. Fig. 7.5 exhibits the RMSE behavior according to the n_a value. Values of n_a greater than 10 do not decrease significantly the RMSE rate, thus justifying the choice of $n_a = 10$. A unique value of n_a is necessary for all the signals measured so that the dimension of $\boldsymbol{\theta}$ is the same as required for the supervised learning method.

In the raw dataset case, the nominal and faulty signals are alike which makes the ideal n_a similar. R^2 values for each dataset are reported in Table 7.1. The tendency is a mean value of R^2 close to unity for these cases. On the other hand, in the cases where baseline subtraction is applied, the defect signals are not similar to the nominal ones. The application of the same n_a for each of the samples results then in different values of R^2 among every sample. In fact, that implies a lower mean of R^2.

Table 7.1: Mean R^2 values for each dataset for $n_a = 10$ when using AR models. Notice that the raw datasets have values close to 1

Data	**Mean $\mathbf{R^2}$**
60 kHz baseline subtracted	0.5397
60 kHz raw	0.9987
140 kHz baseline subtracted	0.3517
140 kHz raw	0.9971
240 kHz baseline subtracted	0.4761
240 kHz raw	0.9994

The latter discussion is reflected in the visualization of the $\boldsymbol{\theta_i}$ (i related to each of the channels) values, for instance for the *raw* and the *baseline subtracted* 60 kHz cases which are depicted in Fig. 7.6. Notice that in Fig. 7.6(a) nominal and defect cases are somehow distinguishable while in Fig. 7.6(b) the features are similar.

In this paper, all the $\boldsymbol{\theta}$ from the NAR models were calculated utilizing the same n_a selected for the AR models. The nonlinearity order was set as $l = 2$, which contains second-order polynomial terms and also the respective linear models. After considering the chosen model order n_a, one can analyze the results of each database.

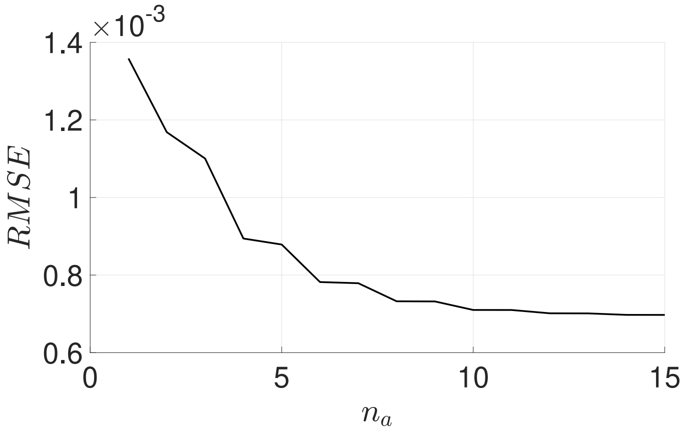


Figure 7.5: RMSE values as a function of n_a. As can be seen, n_a values greater than 10 do not decrease the values of RMSE substantially, therefore, justifying the choice of $n_a = 10$.

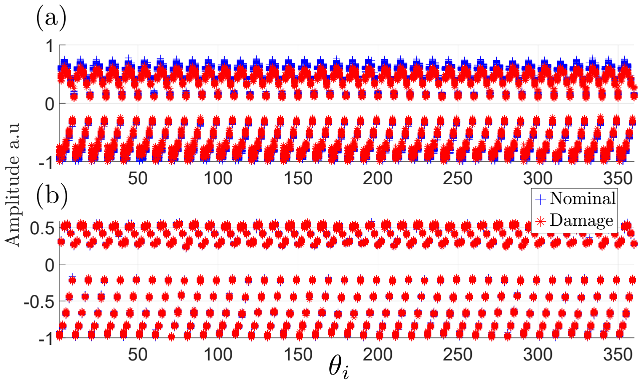


Figure 7.6: $\boldsymbol{\theta_i}$ values of the (a) *60 kHz baseline subtracted* and (b) *60 kHz raw* cases. The features of the nominal samples are in blue while the damage cases are in red. Notice how the classes are distinguishable for the *baseline subtracted* case. It is interesting to mention that the size of the $\boldsymbol{\theta_i}$ is related to each of the channels with $\boldsymbol{n_a}$ equal to 10. There are 36 channels, therefore, the total number of features concatenated is 360.

7.3.2 Classification Results

The mean and best accuracy of all the datasets with AR and NAR feature extraction are summarized in Table 7.2. Observing the AR models for feature extraction, all the architectures investigated are able to accomplish a flawless classification in the *60 kHz baseline subtracted* and *140 kHz baseline subtracted* dataset. For a higher frequency, such as 240 kHz, at least two different architectures of ANNs were also able to achieve a perfect classification. In the *raw* cases, a tendency is noted, the greater the frequency, the better the best accuracies: 0.6500, 0.6667, and 0.7167, for the 60 kHz, 140 kHz, and 240 kHz respectively. The mean accuracy of all *raw* datasets ranges between 0.52 and 0.62 which is nearly equivalent to a heads or tails, that is, equal to 0.5.

In order to obtain better results with the raw datasets, NAR feature extraction is also considered. Most of the *baseline subtracted* datasets results remain approximately equal. All the frequencies obtained the best accuracies of 1.0000 in at least three architectures of the network. For the *60 kHz raw* and *140 kHz raw*, the best accuracy also remains the same. However, for a higher frequency dataset, like the *240 kHz raw*, NAR provided improvements: best accuracy of 0.75. This performance of the NAR approach indicates that this strategy might be better for higher frequencies raw datasets.

In summary, the results indicate that: (i) in the baseline subtracted case the linear AR models are sufficient to obtain perfect classification results as the LOOCV shows for all frequencies tested, what is an important result considering that the dataset is obtained using a real-world experiment and the model has been cross-validated; (ii) nonlinear feature extraction methods such as NAR were able to improve the accuracy in the baseline-free case, which is a more challenging scenario. This can be important for future developments where baseline signals are not available, which is currently the case in many applications.

Table 7.2: Results of mean and best accuracy for all frequencies and the different architectures for ANN considering LOOCV strategy. The AR feature extraction considering baseline subtraction achieves perfect accuracy in all frequencies analyzed. The NAR feature extraction method conserves or improves the results for the raw datasets.

Architecture	60 kHz raw		60 kHz baseline subtracted		140 kHz raw		140 kHz baseline subtracted		240 kHz raw		240 kHz baseline subtracted	
	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best
	AR Models											
{10}	0.5556	0.6167	0.9994	**1.0000**	0.5439	0.6000	0.9933	**1.0000**	0.6106	0.6833	0.9828	0.9833
{100}	0.5406	**0.6500**	0.9994	**1.0000**	0.5183	0.6000	0.9744	**1.0000**	0.5789	0.6500	0.9806	0.9833
{10 10}	0.5444	0.6333	0.9994	**1.0000**	0.5367	0.6333	0.9917	**1.0000**	0.6194	0.6833	0.9822	0.9833
{100 100}	0.5222	0.5833	0.9983	**1.0000**	0.5244	**0.6667**	0.9822	**1.0000**	0.5944	0.6667	0.9822	0.9833
{10 10 10}	0.5439	0.6167	0.9994	**1.0000**	0.5339	0.6333	0.9906	**1.0000**	0.6061	0.6833	0.9839	**1.0000**
{100 100 100}	0.5367	**0.6500**	0.9983	**1.0000**	0.5244	0.6167	0.9806	**1.0000**	0.5883	**0.7167**	0.9811	**1.0000**
Architecture	NAR Models											
{10}	0.5317	**0.6333**	0.9994	**1.0000**	0.5144	0.5833	0.9861	**1.0000**	0.6650	**0.7500**	0.9817	**1.0000**
{100}	0.5206	**0.6333**	0.9839	**1.0000**	0.4967	0.5833	0.9111	0.9833	0.6194	0.7000	0.9689	**1.0000**
{10 10}	0.5367	0.6167	0.9944	**1.0000**	0.5150	**0.6500**	0.9844	**1.0000**	0.6222	0.7000	0.9811	**1.0000**
{100 100}	0.5172	**0.6333**	0.9872	**1.0000**	0.4989	0.6000	0.9367	0.9833	0.6017	0.6833	0.9644	0.9833
{10 10 10}	0.5200	**0.6333**	0.9972	**1.0000**	0.5172	0.6333	0.9728	**1.0000**	0.6067	0.6833	0.9783	**1.0000**
{100 100 100}	0.5122	0.6000	0.9906	**1.0000**	0.4972	0.6000	0.9322	0.9833	0.5811	0.6667	0.9700	0.9833

8 Machine Learning Based Corrosion-like Defect Estimation with Shear-Horizontal Guided Waves Improved by Mode Separation

In the current chapter, the use of the first two SH GW modes for estimation and detection of corrosion-like defects in aluminium plates is investigated. The main contribution herein is to show that mode mixing removal improves the effectiveness of ML predictive models. Numerical simulations have been performed to generate time series for creating the estimators, while experimental data have been used to validate them. We show that a full mode separation scheme decreased the error rate of the final model by 30% and 67% in defect estimation and detection respectively.

8.1 Problem Description

In this case study, signals due to the interaction of SH GW in an aluminium plate with wall thinning defects were used to extract features for a ML model, used to estimate the corrosion-like defect severity. The numerical and experimental data herein used were pre-acquired by other member and released to this work. Details on the data acquisition are reported in [57]. These data were previously used in [56, 57, 32]. Numerical data served to train the model while experimental data were used to validate it. The setup follows [32, 57]. The geometry of the plate can be observed in Fig. 8.1. The plate is 8-mm thick with a 150 mm long thinning region starting at 182 mm from the origin, where transmission occurs. Either its depth (d) and edge angle (α) were varied. Reception is performed either before the defect or after it, positions (1) or (2) in Fig. 8.1, respectively.

Numerical data were obtained by means of a commercial, time-domain, Finite Element Method (FEM) solver, PZFlex©. Simulations were performed in a two-dimension model of a plate similar to the one used in the experimental setup as shown in Fig. 8.1. The material density was set to $2698\,kg/m^3$, and the transverse wave speed equal to $3111\,m/s$. Due to variation of α and d, the dataset created has approximately 1230 defect cases, which includes the simulated signals of papers [32, 57] and extend with more signals to allow

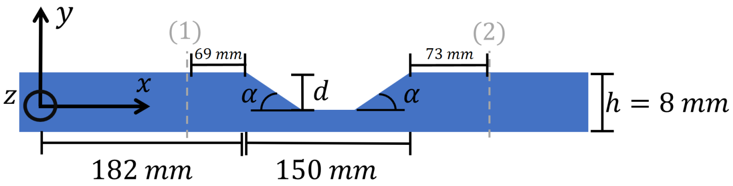


Figure 8.1: Representation of the aluminium plate and defect geometry. Generation is imposed at the origin and reception either before and after the defect, position (1) and (2) respectively. The corrosion severity is evaluated based on the relative area of the defect which is dependent of d and α.

separated and non-separated modes to be available to the construction of the ML model.

Experimental data were acquired through the experimental setup used in [56, 32]. Using a RITEC ©RPR4000 Pulser/Receiver to generate and receive the signals from PPM EMATs, used as transmitter and receiver. Both emitter and receiver EMATs were 3 cycles 10 mm nominal wavelength EMATs from Sonemat Ltd. Aluminium plates of 800 mm × 250mm and 8 mm thickness were used. The experiment was carried with 34 samples, one instance is the nominal condition (without defect) and other 33 samples with variations of α (10°, 45°, and 90°) and d (1mm, 2mm, 3mm,..., 7mm), with $\alpha = 55°$ varying d (2mm, 3mm,..., 7mm), plus six additional samples, namely, d =6 mm at $\alpha =$ 25°, 30° and 35° and d = 7 mm at α = 25°, 30° and 65°. It is interesting to mention that the experiment was performed without specifically controlling the environmental conditions such as temperature and humidity, therefore, making the defect estimation problem even harder. The excitation pulse was set to an 8 cycle tone burst at either 311 kHz or 367 kHz, in order to predominantly generate the SH0 or SH1 modes at a 10 mm nominal wavelength, respectively.

An interesting aspect of the experiment is the fact that the signals were acquired in different transmitter-receiver configurations. Based on Fig. 8.1, the transmitter longitudinal position is fixed at the origin, however, it is positioned either at the upper as well at the lower surfaces of the plate. The receiver is positioned in four different places. Before the defect, (1), both at the upper and lower surfaces; and, after the defect, (2), also, positioned both at the top and then on the bottom surfaces of the plate. Therefore, for each generating frequency and receiving position there are four different signals acquired. Holders were used to ensure alignment between upper and lower transducers which is important to achieve the mode separation technique employed in [56].

The excitation pulse was set to an 8 cycle tone burst at either 311 kHz or 367 kHz, in order to predominantly generate the SH0 or SH1 modes at a

10 mm nominal wavelength, respectively. Both modes can be generated and received in this case, we assess whether mode separation further improves the ML model accuracy, utilized either in transmission or reception, as explained in the following section. More details about the experiment setup as well as the simulation procedure can be found in detail in [32, 56].

8.2 Method

In the present section, we state all the methods used in the overall data-driven modeling construction pipeline. Fig. 8.2 summarizes all the steps performed for the creation of the predictive models for this case study. Below we describe each of these steps, together with a description of what was defined in the present work with justifications.

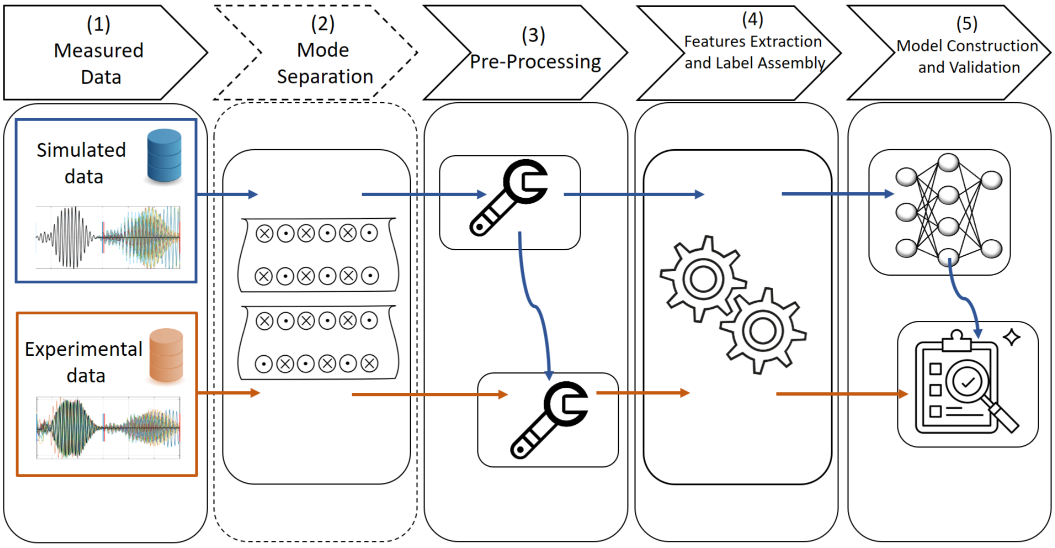


Figure 8.2: Overview of the modeling workflow solution adopted. Note that the simulated data is used in order to create the model, while the experiments to validate it. Such workflow is general and may be replicated in other cases where high-fidelity computational models are available and a number of cases can be computed in a feasible time.

Block number (1) contemplates all the measured data exploited in this work. Data from experiments and simulations are used. The former is used to validate the model, while the latter is used to create it. In the present case study, we show that it is possible to create a data-driven model using simulated data that can be validated with experiments. The present approach is important as in many cases data is expensive to measure, making damage estimation with the use of ML non-viable as it is a scenario which demands a reasonable amount of data to be accomplished in an efficient way.

The second stage of the pipeline is the block number (2) of Fig. 8.2, entitled *Mode Separation*. Given the 8mm thickness of the current plate, signals available in their raw format, namely with no mode separation pre-processing, may be subject to the presence of many modes. In this case, this happens due to two main reasons. (i) Excitation of a single mode is impractical due to the number of spatial cycles in the PPM array which is finite. This results in a finite frequency bandwidth so, even if the proper frequency and wavelength to generate a specific mode are selected [96], other SH modes can be generated if their dispersion curves cross the transducer operating region[27, 32]. Fig. 8.3 shows the -6 dB border of the operating regions of a 3 cycle PPM EMAT, driven by an 8 cycle tone-burst at 311 kHz and 367 kHz. As can be seen, in this case, when attempting to generate the SH0 mode, the SH1 mode is also generated. (ii) Depending on the mode which impinges upon the corrosion-like defect there is a phenomenon called mode conversion in either reflection or transmission of the defect [96, 56]. The latter can happen with a signal already in the presence of multi-modes which can result in an even more complex signal [97]. Therefore, rendering defect sizing difficult. In order to solve this problem, we can adopt a mode separation scheme [98, 99, 100, 101]. Herein, we adopted dual transduction following the procedure proposed in [56, 102]. In a nutshell, it consists of positioning transmitter transducers on both the upper and lower surfaces of the plate at the same longitudinal position. If the forces imposed by both transducers are in-phase, resp. out-of-phase, then only symmetric, resp. anti-symmetric modes are generated.

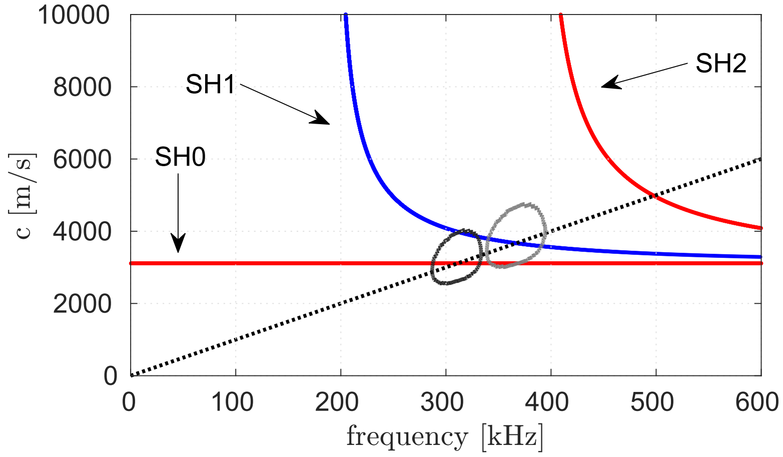


Figure 8.3: Phase speed dispersion curves for SH GW in an 8 mm thick aluminium plate. The dotted line means a 10 mm nominal wavelength. The continuous closed lines represent the -6 dB border of the operating region of 3 a cycle PPM EMAT driven by an 8 cycle tone-burst at 311 kHz (black line) and 367 kHz (gray line).

In order to illustrate the mode mixing, Fig. 8.4 shows the propagation of mode SH1 when interaction with a 3 mm deep defect with a taper angle of 50°. In this case, a pure SH1 mode was generated, adopting the mode-generation in transmission scheme, even so, mode mixing does occur, due to mode–conversion, either in reflection or transmission, as can be clearly observed. This exemplifies the complex behavior when working in the high frequency-thickness regime.

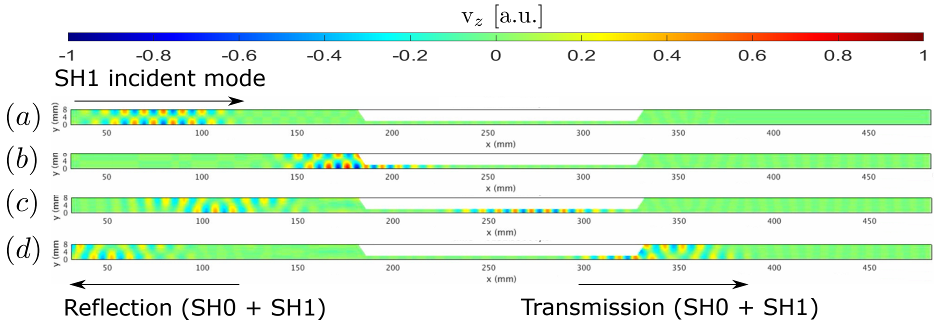


Figure 8.4: Propagation of the SH1 particle velocity wavefield and mode conversion phenomenon in several time instants. At 40.95 μs (a), it can be clearly seen the anti-symmetric wave structure of the incident SH1 mode. At 79.45 μs (b), when facing the first defect edge, the incident SH1 mode is partially reflected and transmitted to the defect. The reflected filed is composed of SH0 mode, generated by mode-conversion as well as the SH1 mode itself.. At 109.95 μs, (c) the propagation proceeds, and the clear anti-symmetrical mode pattern, which was previously observed, is no longer identified in the reflected waves due to mode mixing between the SH0 and the SH1 modes (SH0+SH1), whilst within the defect are there is clearly only the symmetric pattern of the SH0 mode since the remaining thickness in the case is below the cut-off thickness of the SH1 mode. At 132.95 μs, (d), another mode-conversion arises at the right end of the defect, this time from the SH0 mode to the SH1, thus showing a similar mode-mixing as in reflection. Adapted from [103]

The methodology was implemented in three different ways, namely, (i) *fully mode-separation*, (ii) *mode-separation in transmission*, and (iii) *non-separated*. In the latter, there is no separation at all and the *Mode Separation* block can simply be skipped. Each mode separation methodology can be summarized as follows:

i In *fully mode-separation* a high purity SH0 or SH1 is generated at its respective optimum frequency, then mode-separation is performed in reception in order to obtain the converted and non-converted modes. This means that for a single generation, for instance, of the SH0 mode, there are two sets of received signals, at each position, one for the received SH0 mode and another for the received SH1 mode, obtained due to mode conversion. All mode-mixing is eliminated in this case.

ii In *mode-separation in transmission* a pure mode is generated, but no mode separation is performed in reception. This means that the received signal may present mode mixing due to mode conversion, but not due to two generated modes.

iii Lastly, *non-separated* means that there is just one transmitter and one receiver positioned only on one surface of the plate. In this case, there is mode-mixing in transmission and in reception.

Fig. 8.5 to Fig. 8.10 present the signals resulted when applying the different mode-separation schemes. We only reported the signals generated with the 367 kHz frequency. A similar set of signals can be plotted with the 311 kHz frequency which is also explored in this work, however, they are not shown for the sake of brevity. The black line with greater width indicates the non-defective plate and the different grey-scaled lines indicate signals obtained for different defects severities, a lighter color meaning a more severe defect (greater relative area), a darker color meaning a less severe defect, the color-severity relation is described by the color bar. The excitation pulse of each case is defined within the blue vertical lines while the reflected and transmitted waves are defined by the red vertical lines. These time gates are defined based on the expected arrival time of both the reflection and transmission signals according to their group speed. These are strategically defined in order to avoid any interference such as reflections at the plate's end. It is worth to mention that only the simulated signals that have an experimental counterpart were displayed in Fig. 8.5, Fig. 8.7, and Fig. 8.9 to facilitate visualization and comparison.

The *fully mode-separated* signals for a purely generated SH1 are shown in Fig. 8.5 and Fig. 8.6 for numerically simulated signals and experimental signals respectively. In subplots (a) and (c) at around $90\,\mu s$ we can observe the reflected signals. In subplots (b) and (d) the transmitted signals are shown. Notice that subplots (a) and (b) represent signals with pure SH1 modes while (c) and (d) are composed exclusively of SH0 mode signals. When comparing experimental and simulated signals, we can notice that the excitation pulse of the experimental cases has slight phase differences from each other due to the experimental positioning errors. One can also observe that for early times, the excitation pulse is affected by interference from the high-power excitation signal applied to the EMAT, which is not present in the numerical simulation. Both simulated and experimental signals present different amplitudes and phase shifts in reflection and transmission which are inherent to the different geometries of each defect.

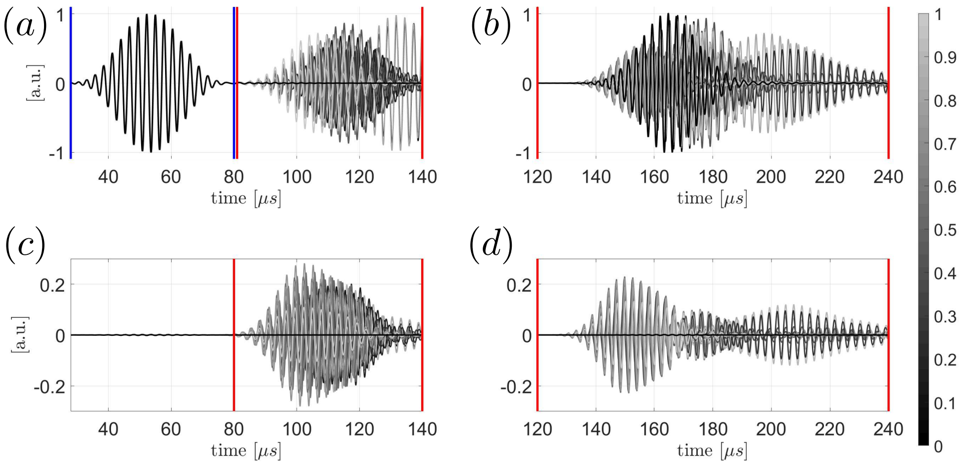


Figure 8.5: Simulated *fully mode-separation* signals for purely SH1 generated at 367 kHz. Reception of the separated SH1 mode (a) and (b) and separated SH0 mode (c) and (d). Reception before the defect, at position (1), (a) and (c) and after the defect, position (2), (b) and (d).

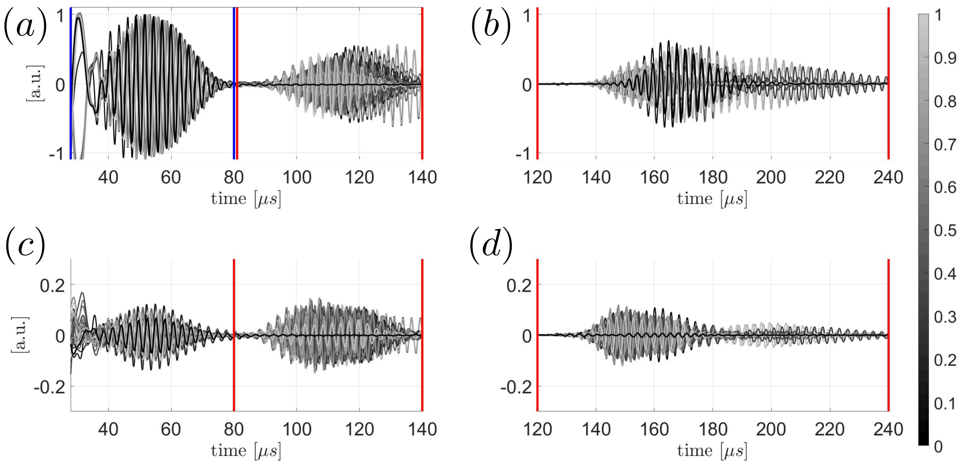


Figure 8.6: Experimental *fully mode-separation* signals for purely SH1 generated at 367 kHz. Reception of the separated SH1 mode (a) and (b) and separated SH0 mode (c) and (d). Reception before the defect, at position (1), (a) and (c) and after the defect, position (2), (b) and (d).

Mode-separation in transmission signals are shown in Fig. 8.7 for simulation and Fig. 8.8 for experiments. In this case, a high purity SH1 mode is still generated, but the reflected and transmitted signals have both SH0 and SH1 modes which interfere. Finally, *non-separated modes* signals are shown in Fig. 8.9 and Fig. 8.10 for simulation and experiments, respectively. As can be seen, the direct pulse differs from previous cases, since the SH1 is not purely generated. It is worth noticing that either *non-separated modes* or *mode-separation in transmission* results in a single received signal at each position, related to the response signal at one surface of the plate. In *fully mode-separation*, however, two signals were obtained for each generation due to the separation in reception which is performed through the sum of response on the upper and lower surfaces of the plate.

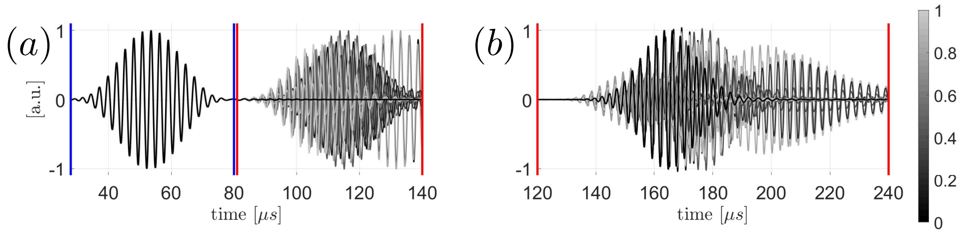


Figure 8.7: Simulated *mode-separation in transmission* signals generated at 367 kHz. Reception before the defect, at position (1), (a) and after the defect, position (2), (b). Note that the *mode-separation in transmission* signals result in a single received signal at each position.

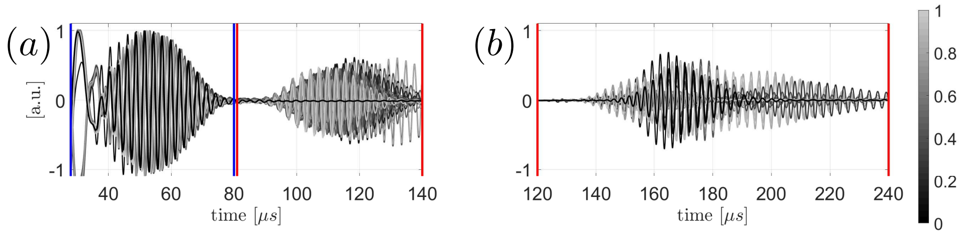


Figure 8.8: Experimental *mode-separation in transmission* signals generated at 367 kHz. Reception before the defect, at position (1), (a) and after the defect, position (2), (b). Once again, as previously mentioned, the procedure of mode separation only at the transmission generates a single received signal.

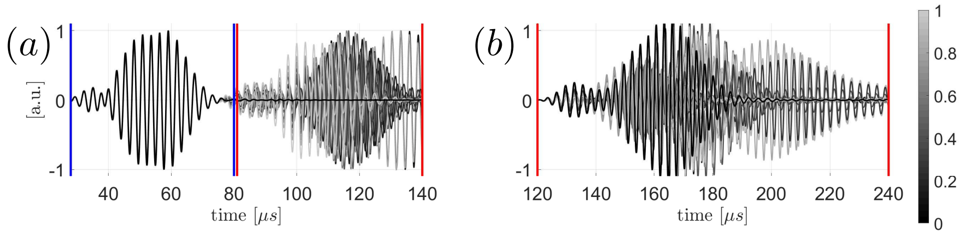


Figure 8.9: Simulated *non-separated modes* signals generated at 367 kHz. Reception before the defect, at position (1), (a) and after the defect, position (2), (b). No mode separation is performed, therefore, a single signal is generated in reflection and transmission for the attempted mode simulated.

Continuing the step-by-step methods of this work depicted in Fig. 8.2, we reach block number (3). The raw experimental data should be pre-processed in order to produce valid results when compared to simulated signals. In order to

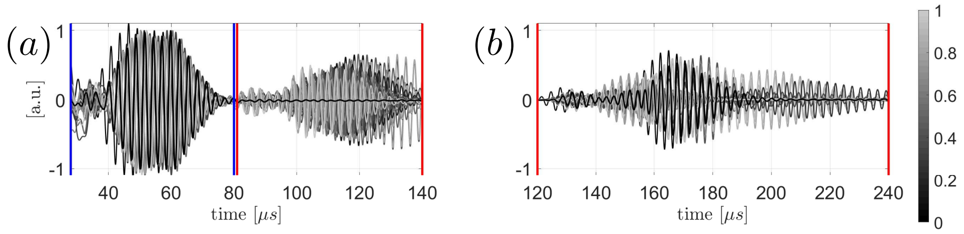


Figure 8.10: Experimental *non-separated modes* signals generated at 367 kHz. Reception before the defect, at position (1), (a) and after the defect, position (2), (b). Because no mode-separation was performed in these experimental data, there is only a single signal for reflection and transmission.

assess a fair comparison in the process of supervised learning models testing, we adopt data normalization and correct time window, like the following:

- Normalization: Considering linearity, we multiplied the measurements by the inverse of the maximum absolute amplitude of each incident wave. The amplitude of experimental and simulated signals differ mainly due to attenuation, which was not introduced in the numerical simulation. Therefore, the amplitude of the experimental signals was also corrected based on the same case of the simulated wave signal within the proper time windows.
- Define correct time window: We define the time window that contains relevant information about the transmission and reflection measurements. The goal is to use the time history most relevant to the phenomenon at hand.

After the pre-processing of data, the feature extraction and label assembly stage is performed, block number (4), Fig. 8.2. A feature extraction process is necessary to reduce the high-dimensional of time series in a way that the model construction process is possible. The goal is to obtain features that are sensitive to the defect parameters and can be representative of the physical characteristics received signals in a reduced dimension when compared to the full amount of samples of the time series, thus subsequently used as input for the model [20]. In the present work, we focus on temporal and statistical features which are summarized in Table 8.1.

The number of total features depends on each of the cases (i)-(iii) for mode separation. In case (i), 8 patches (transmitted or receiving signals within the red vertical lines of Fig. 8.5 and Fig. 8.6) are available, 4 for the signals generated with 311 kHz and 4 for the signals generated for the 367 kHz. Therefore, the L_2-energy norm, the Curve length, the Kurtosis, and the self-explanatory amplitude related (7 features) are extracted from the 8 patches of signals which results in 80 features. Besides those, the transmission and reflection coefficients culminate in 8 new features. All of that added to the two Pearson correlation coefficients yields a total of 90 features. All the features which were previously extracted from 8 patches in case (i) are obtained for only 4 patches in case (ii) and case (iii), resulting in 40 features in each case. Recall that case (i) results in two signals for each generation in each position, whereas cases (ii) and (iii), result in just one signal at each position, thus presenting a different number of patches. Reflection and transmission coefficients add 4 more features that, together with the two Pearson correlation coefficients, sums up to 46 features.

Table 8.1: Description of the temporal and statistical features extracted from the (i)-(iii) strategies of mode separation. By using simple feature engineering we are able to provide fast yet informative solutions for supervised learning.

Feature	Description
1	L2-energy norm (see Section 4.2)
2	Curve length (see Section 4.2)
3	Kurtosis (see Section 4.2)
4	Maximum amplitude
5	Number of peaks greater than 60% of the maximum amplitude
6	Number of peaks greater than 20% of the maximum amplitude
7	Mean of peak values greater than 60% of the maximum amplitude
8	Mean of peak values greater than 20% of the maximum amplitude
9	Standard deviation of peak values greater than 60% of the maximum amplitude
10	Standard deviation of peak values greater than 20% of the maximum amplitude
11	Reflection Coefficient: ratio of the amplitudes of the reflected to the incident waves at the position (1) in Fig. 8.1.
12	Transmission Coefficient: ratio of the amplitudes of the wave transmitted through the whole region of the defect to the incident wave at position (2) and (1), respectively, in Fig. 8.1.
13	Pearson Coefficient: correlation of the incident and reflected wave (see Section 4.2)

The target is also defined in this stage. We set the target of the model as the damage indices to be predicted. As the definition of whether there is a defect or not may be inferred by sole visualization of the SH ultrasonic GW response field, the same cannot be stated in such a specific manner to the severity of the defect. The severity of the defect is measured by the relative area of the thinned defect. Namely, in Fig. 8.1, the severity of the defect will be given by the area of the trapezoid that varies as a function of α and d relative to a full defect in the plate.

We perform two types of model in the present work, on the basis of the area of the defect:

- Regression: the area of the defects are normalized between 0 and 1, which correspond to the minimum and maximum area, or the nominal condition and largest area of the defect-like corrosion respectively. The normalized values are then used as the target variable.
- Classification: the classification models separate the area of the defects into three regions, namely low (relative area < 0.3), medium (relative area ≥ 0.3 and < 0.6), and high severity (relative area ≥ 0.6).

Finally, in the last block of Fig. 8.2, block (5), we bring into play the models construction and validation. The extracted features and respective labels representing damage indices (for regression and classification) can be viewed as input-output pairs, which can be employed by any supervised learning, in order to solve the defect estimation problem. In the present work, we solved the regression and classification ML problems with linear models (LM) and multilayer perceptron artificial neural networks (MLP), both of these models can be referred to in subsections 5.1 and 5.2. Both models were trained with approximately 1230 simulated signals and validated with the 34 experimental signals. In order to compare different mode separation strategies for feature extraction, we adopted different performance metrics that can evaluate both the classification and regression problems. These are accuracy, Kappa coefficient, coefficient of determination (R^2), and area under accuracy curve (AUAC). For a better understanding of the aforementioned metrics, please refer to Section 6.3.

8.3 Results

In the present section, we devise the results when applying classification and regression-based supervised learning with linear models and artificial neural networks, to the problem of defect estimation using SH GW (Section 8.1). The goal is to evaluate the mode separation strategy to time-based feature extraction methods in order to assess its improvements.

The linear model was first adopted in order to check if a simple linear solution could successfully solve the problem and so assess the difficulty of the damage estimation problem. Since its performance was not satisfactory, non-linear models with greater prediction capability such as artificial neural networks were adopted as well and turned out to dramatically improve the results. We tested different architectures for the artificial neural networks, a strategy to tune hyperparameters of the model, varying the number of hidden layers, and the number of neurons in each of these layers. The number of hidden layers is set from 1 ({X}) to 3 ({X X X}) and the number of neurons in each of these layers as 10 (X=10), 50 (X=50), or 100 (X=100). The activation function of the artificial neural networks is the hyperbolic tangent in the hidden layers. The output layer for the classification model is a softmax, while for the regression a linear layer. After some trial and error, the scaled conjugate gradient backpropagation (SCG) learning algorithm is adopted. In order to statistically evaluate the results presented in this section, all the supervised learning models were executed 33 times with different random seeds.

The results are compared for the three mode separation schemes: (i) *fully mode-separation*, (ii) *mode-separation in transmission*, and (iii) *non-separated* signals as detailed in Section 8.1 for feature extraction. The presentation of the results will be divided into classification and the regression-based solutions, which will be explored respectively in Subsections 8.3.1 and 8.3.2. It is important to mention that all results herein reported have been tested with experimental data, which is unseen in the estimation of the black-box model. We thus use hold-out cross-validation. The training data was generated by a series of FEM simulations, in every model construction, 85 % of the data were used to train while 15 % to validate the model.

8.3.1 Classification Results

The resolution problem through classification with linear and MLP models was divided according to the mode separation consideration. Table 8.2 reports the results. In case (iii), results achieved by the MLPs are satisfactory, with mean accuracy ranging from 0.73 to 0.81 and best accuracy reaching 0.9118. Results of the case (ii) are also shown. The mean accuracy dropped to a range between 0.69 and 0.80, and the best accuracy remained 0.9118. The latter was not able to improve the results of MLP models, compared to the case (iii). Interestingly, outcomes, when modes are separated in reception, produced the best results, that is after interaction with a defect, case (i). This indicates that the mode conversion phenomenon, and the consequent mode-mixing of converted modes, indeed produce complicated signals in the received signal, which when separated facilitates analysis. In this case, the best accuracy was actually 0.9706, when the classifier made only one single misclassification in the test dataset. This result is quite important, showcasing the advantage of performing mode separation for feature extraction. The range of mean accuracy also improves, varying from 0.79 to 0.85. Concerning the tuning of the architecture of models, no specific number of neurons or number of layers generally obtains the best results. However, according to the best case, (i), a single layer with 10 neurons, and an MLP with three hidden layers with 50 or 100 neurons are the best architectures to adopt, obtaining the best mean accuracy and the best single accuracy respectively. The train results are mostly greater than the test results in all cases. This is expected since the data used is in fact the same utilized to train the model, that is, the simulated signals.

Lastly, regarding the Kappa coefficient, all the models showed at least, reasonable results. All results were greater than 0.5, which validates the fairness of the models. Looking at the linear model accuracy, one can infer that the results are not satisfactory when compared to the nonlinear models based on MLPs, regardless of the pre-processing cases (i)-(iii). We kept the results of the linear model, to compare the MLP to a baseline and showcase the difficulty of the present case study. In the following, we check the results of the regression models for defect estimation.

8.3.2 Regression Results

Concerning the results achieved by the regression models, the outcome is reported in Table 8.3. The case (iii) presents the best R^2 and AUAC, respectively, as 0.9089 and 0.9356, while the mean AUAC falls in a range from 0.84 to 0.90. The architecture has a major role in the mean R^2, as different MLPs imply a higher standard deviation for this metric. Case (ii) reports results slightly different from case (iii). The best R^2 is 0.8953 and the best AUAC corresponds to 0.9285. In case (i) signals, results are far superior to the ones attained by (ii) and (iii), especially when we address the mean results of AUC and R^2. Results show, once again, that the mode separation procedure facilitates the supervised learning action. The mean R^2 achieved 0.8196 while the mean AUC reached 0.9222. The best AUAC of the signals is equivalent to 0.9486 and the best R^2 is equal to 0.9211. The linear regression obtained poor results when compared to the ones originated from the MLPs regardless of the mode separation. When we look at the architecture hyperparameters, it is noticed that different specific numbers of neurons and hidden layers are the most adequate for each case, however, the best results are obtained with case (i), an MLP with 100 neurons jointly with 3 hidden layers. The train results are greater than the test ones which is expected. The same occurred in the aforementioned classification results.

Highlighting the R^2 metric, one can notice its performance in Fig. 8.11.

Table 8.2: Results in terms of accuracy and Kappa metric for all the classification models tested using the data pre-processing cases (i)-(iii). The improvement with respect to the error rate of the nonlinear models compared to the linear ones are 90%, 63%, and 67% for the cases (i)-(iii), respectively. When we compare the best models for each mode separation cases (i)-(iii), we have for the error rate 67% improvement in case (i). We can clearly see by the results that the nonlinear models with full mode separation strategy perform better.

Case	Model { Architecture }	Accuracy (train)	Accuracy (test)	Kappa (train)	Kappa (test)	Best accuracy
(i)	MLP {10}	0.9325 ± 0.0177	0.8538 ± 0.0399	0.9010 ± 0.0252	0.7984 ± 0.0699	0.9412
	MLP {50}	0.9219 ± 0.0776	0.8244 ± 0.0800	0.8840 ± 0.1174	0.7526 ± 0.1270	0.9118
	MLP {100}	0.8744 ± 0.1516	0.7888 ± 0.1135	0.8125 ± 0.2287	0.6918 ± 0.1728	0.9118
	MLP {10 10}	0.9225 ± 0.0292	0.8262 ± 0.0501	0.8881 ± 0.0375	0.7558 ± 0.0770	0.9412
	MLP {50 50}	0.9340 ± 0.0210	0.8307 ± 0.0713	0.9030 ± 0.0312	0.7589 ± 0.1034	0.9118
	MLP {100 100}	0.8943 ± 0.0999	0.8111 ± 0.1043	0.8431 ± 0.1503	0.7295 ± 0.1504	0.9412
	MLP {10 10 10}	0.9166 ± 0.0905	0.7995 ± 0.1031	0.8855 ± 0.0926	0.7230 ± 0.1129	0.9412
	MLP {50 50 50}	0.9175 ± 0.0593	0.8084 ± 0.1024	0.8779 ± 0.0923	0.7290 ± 0.1421	**0.9706**
	MLP {100 100 100}	0.9385 ± 0.0160	0.8378 ± 0.0575	0.9087 ± 0.0235	0.7633 ± 0.0894	**0.9706**
	Linear Model	0.7353 ± 0.0044	0.7059 ± 0.0000	0.6025 ± 0.0067	0.5157 ± 0.0000	0.7059
(ii)	MLP {10}	0.9292 ± 0.0272	0.7718 ± 0.0754	0.8948 ± 0.0407	0.6586 ± 0.1069	0.8824
	MLP {50}	0.9102 ± 0.0845	0.7228 ± 0.0754	0.8661 ± 0.1285	0.5858 ± 0.1101	0.8529
	MLP {100}	0.8939 ± 0.1147	0.6916 ± 0.1149	0.8409 ± 0.1737	0.5482 ± 0.1706	0.8529
	MLP {10 10}	0.9282 ± 0.0252	0.7522 ± 0.0739	0.8940 ± 0.0369	0.6370 ± 0.1002	**0.9118**
	MLP {50 50}	0.9353 ± 0.0235	0.7433 ± 0.0675	0.9036 ± 0.0348	0.6229 ± 0.0959	0.8824
	MLP {100 100}	0.8779 ± 0.1398	0.6925 ± 0.0987	0.8171 ± 0.2101	0.5411 ± 0.1626	0.8529
	MLP {10 10 10}	0.9172 ± 0.0296	0.7977 ± 0.0689	0.8782 ± 0.0432	0.6934 ± 0.0978	0.8824
	MLP {50 50 50}	0.9220 ± 0.0641	0.7424 ± 0.0694	0.8838 ± 0.0958	0.6171 ± 0.1072	0.8824
	MLP {100 100 100}	0.8863 ± 0.1097	0.7193 ± 0.1125	0.8295 ± 0.1661	0.5829 ± 0.1642	0.8824
	Linear Model	0.7987 ± 0.0039	0.7647 ± 0.0000	0.7021 ± 0.0056	0.6180 ± 0.0000	0.7647
(iii)	MLP {10}	0.9009 ± 0.0471	0.7888 ± 0.0651	0.8532 ± 0.0724	0.6792 ± 0.0862	0.8529
	MLP {50}	0.8985 ± 0.0575	0.8128 ± 0.0514	0.8519 ± 0.0703	0.7162 ± 0.0815	**0.9118**
	MLP {100}	0.8655 ± 0.1185	0.7754 ± 0.0685	0.7996 ± 0.1781	0.6503 ± 0.1163	0.8824
	MLP {10 10}	0.9042 ± 0.0381	0.7977 ± 0.0408	0.8607 ± 0.0553	0.6898 ± 0.0652	0.8529
	MLP {50 50}	0.9204 ± 0.0548	0.7923 ± 0.0674	0.8831 ± 0.0795	0.6841 ± 0.1042	**0.9118**
	MLP {100 100}	0.8744 ± 0.1009	0.7638 ± 0.0905	0.8120 ± 0.1540	0.6422 ± 0.1319	**0.9118**
	MLP {10 10 10}	0.8835 ± 0.0638	0.7754 ± 0.0601	0.8314 ± 0.0911	0.6546 ± 0.0895	0.8824
	MLP {50 50 50}	0.8758 ± 0.0988	0.7308 ± 0.0965	0.8161 ± 0.1483	0.5904 ± 0.1412	0.8529
	MLP {100 100 100}	0.8596 ± 0.1196	0.7389 ± 0.1083	0.7897 ± 0.1804	0.6034 ± 0.1564	0.8824
	Linear Model	0.7969 ± 0.0041	0.7353 ± 0.0000	0.6997 ± 0.0059	0.5591 ± 0.0000	0.7353

The correlation between prediction and true response is seen. The best cases of all the datasets are presented. It is notable how case (i) results, obtained in the {10 10 10} MLP architecture, are closer to the ideal prediction, $R^2 = 1$ which is portrayed by the dashed black line. The closer the data is to the line, the higher the R^2. The AUAC of the best cases of the three datasets are also presented in Fig. 8.12. Notice how the AUAC of (i) encompasses most of the graph, while (ii) and (iii) embrace a lesser area.

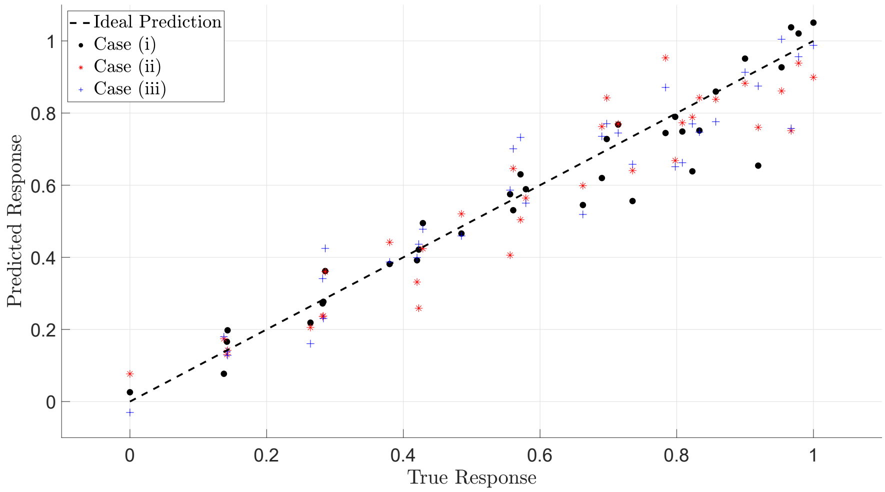


Figure 8.11: Best test results for true versus predicted response. Note that results of case (i), achieved by the MLP { 10 10 10 }, are closer to the dashed line which represents ideal prediction ($R^2 = 1$).

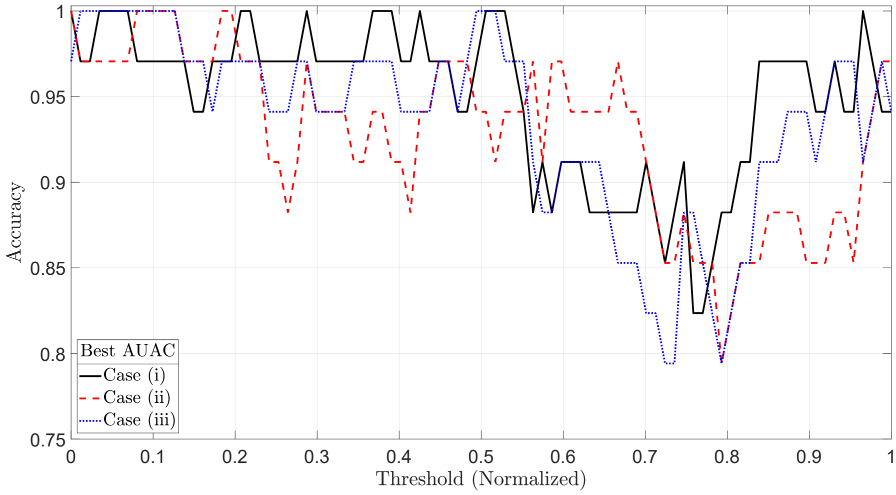


Figure 8.12: Best test results for the AUAC of the best cases from all (i)-(iii) mode separations. One can infer that the curve achieved by case (i) is the one closer to unity.

Finally, the classification results can be compared to the regression results. When comparing the error rate between the classification accuracy and the regression AUAC, it is seen that the case (i) dataset achieves 67% error rate improvement while the error rate of the regression AUAC drops by 30% with the mode separation treatment. In the classification, the data is classified as three different outputs (low, medium, and high severity), whilst regression treats the data as a continuous output. The choice on which approach is used depends on whether fine monitoring must be employed or categorical output is enough. In the former, a regression would be recommended, whereas, in the latter, a classification strategy suffices. The regression strategy did accomplish reasonable results, $R^2 = 0.9211$. Results have shown that a 97.06% accuracy is found using classification with full mode separation.

Table 8.3: Results in terms of R^2 and AUAC for all the regression models tested using the data cases (i)-(iii). Looking at the best R^2 and AUAC, the case (i) obtained the greatest result, 0.9211 and 0.9486 respectively. Once again, another indication that the nonlinear models with full mode separation achieve the greatest performance.

Case	Model { Architecture }	R^2 (train)	R^2 (test)	Best R^2	AUAC (train)	AUAC (test)	Best AUAC
(i)	MLP {10}	0.9783 ± 0.0040	0.7910 ± 0.0710	0.9045	0.9727 ± 0.0031	0.9201 ± 0.0140	0.9456
	MLP {50}	0.9710 ± 0.0155	0.3685 ± 0.4015	0.7667	0.9678 ± 0.0081	0.8645 ± 0.0301	0.9199
	MLP {100}	0.9644 ± 0.0288	-0.2655 ± 0.7682	0.8282	0.9646 ± 0.0114	0.8302 ± 0.0464	0.9128
	MLP {10 10}	0.9787 ± 0.0057	0.7989 ± 0.0935	0.9122	0.9729 ± 0.0040	0.9174 ± 0.0163	0.9435
	MLP {50 50}	0.9751 ± 0.0130	0.2909 ± 0.8652	0.8709	0.9708 ± 0.0080	0.8848 ± 0.0327	0.9356
	MLP {100 100}	0.9567 ± 0.0543	-0.0937 ± 0.8140	0.6625	0.9627 ± 0.0168	0.8432 ± 0.0341	0.9080
	MLP {10 10 10}	0.9792 ± 0.0070	0.8196 ± 0.1029	0.9211	0.9734 ± 0.0051	0.9222 ± 0.0131	**0.9486**
	MLP {50 50 50}	0.9744 ± 0.0110	0.6357 ± 0.3080	0.8944	0.9704 ± 0.0069	0.8978 ± 0.0189	0.9314
	MLP {100 100 100}	0.9676 ± 0.0199	0.3881 ± 0.4419	0.8378	0.9665 ± 0.0100	0.8640 ± 0.0398	0.9160
	Linear Regression	0.9690 ± 0.0007	-3.9921 ± 0.0000	-3.9921	0.9649 ± 0.0004	0.6405 ± 0.0000	0.6405
(ii)	MLP {10}	0.9650 ± 0.0172	0.5248 ± 0.2795	0.7884	0.9644 ± 0.0094	0.8658 ± 0.0357	0.9025
	MLP {50}	0.9625 ± 0.0164	0.3037 ± 0.2094	0.7267	0.9629 ± 0.0086	0.8438 ± 0.0275	0.8818
	MLP {100}	0.9596 ± 0.0318	-0.2161 ± 0.4026	0.5513	0.9627 ± 0.0129	0.8103 ± 0.0375	0.8668
	MLP {10 10}	0.9710 ± 0.0137	0.5873 ± 0.1225	0.8953	0.9681 ± 0.0076	0.8759 ± 0.0196	**0.9285**
	MLP {50 50}	0.9691 ± 0.0237	0.3726 ± 0.1994	0.7022	0.9672 ± 0.0106	0.8617 ± 0.0251	0.8940
	MLP {100 100}	0.9642 ± 0.0159	-0.1449 ± 0.5016	0.6656	0.9637 ± 0.0085	0.8185 ± 0.0405	0.8844
	MLP {10 10 10}	0.9698 ± 0.0149	0.6079 ± 0.1276	0.7939	0.9675 ± 0.0078	0.8823 ± 0.0171	0.9131
	MLP {50 50 50}	0.9611 ± 0.0308	0.4563 ± 0.2268	0.7824	0.9630 ± 0.0119	0.8621 ± 0.0271	0.9062
	MLP {100 100 100}	0.9613 ± 0.0446	0.1292 ± 0.4091	0.6869	0.9642 ± 0.0149	0.8426 ± 0.0304	0.8861
	Linear Regression	0.9435 ± 0.0014	0.4404 ± 0.0000	0.4404	0.9520 ± 0.0005	0.8218 ± 0.0000	0.8218
(iii)	MLP {10}	0.9618 ± 0.0153	0.7590 ± 0.0864	0.8692	0.9635 ± 0.0077	0.8999 ± 0.0169	0.9266
	MLP {50}	0.9625 ± 0.0198	0.5767 ± 0.2197	0.8541	0.9636 ± 0.0097	0.8726 ± 0.0252	0.9168
	MLP {100}	0.9515 ± 0.0292	0.2572 ± 0.2838	0.7312	0.9585 ± 0.0124	0.8354 ± 0.0288	0.8940
	MLP {10 10}	0.9579 ± 0.0289	0.7598 ± 0.0732	0.9089	0.9629 ± 0.0133	0.9019 ± 0.0122	**0.9356**
	MLP {50 50}	0.9700 ± 0.0120	0.5888 ± 0.2796	0.8338	0.9678 ± 0.0068	0.8767 ± 0.0261	0.9155
	MLP {100 100}	0.9564 ± 0.0312	0.1509 ± 0.6388	0.6811	0.9610 ± 0.0115	0.8475 ± 0.0294	0.8905
	MLP {10 10 10}	0.9576 ± 0.0265	0.7383 ± 0.0946	0.8692	0.9629 ± 0.0125	0.8974 ± 0.0180	0.9238
	MLP {50 50 50}	0.9647 ± 0.0253	0.6582 ± 0.1299	0.8706	0.9659 ± 0.0115	0.8864 ± 0.0156	0.9231
	MLP {100 100 100}	0.9644 ± 0.0140	0.5295 ± 0.1653	0.7362	0.9642 ± 0.0075	0.8628 ± 0.0212	0.8918
	Linear Regression	0.9401 ± 0.0012	0.5558 ± 0.0000	0.5558	0.9515 ± 0.0005	0.8450 ± 0.0000	0.8450

Part IV

Conclusions

9
Final Considerations

The present work deals with defect evaluation problems based on the employment of ultrasonic GW aided by data-driven ML models. Simulated and experimental signals were explored, different feature extraction methodologies were adopted, and, ML models such as ANNs were also used. More specifically, two case studies were explored: (i) a SHM strategy to detect flaws in CFRP plates, in this task feature extraction based on system identification techniques are explored and datasets which consider baseline subtraction are investigated; (ii) interaction of SH GW with corrosion-like defects in aluminium plates, being the main concern of the investigation the effect or influence of mode-separation in the faults estimation using ML.

Concerning the composite plate problem, the study coped with an SHM methodology for fault detection in CFRP plates through the use of ultrasonic GW jointly with a system identification strategy in the features extraction process and ANNs, with and without baseline subtraction. The methods proved to be useful in situations where the response to damage is not evident. Results demonstrate that the baseline consideration makes the process of fault detection possible. Perfect results were achieved in all frequencies analyzed when considering LOOCV and baseline subtracted signals. Regarding the raw signals, that is, without subtraction from the baseline, the NAR approach was able to improve the results for the $240kHz$ excitation cases. The NAR approach was able to decrease the error rate results in 13% for the $240kHz$ excitation cases. This may be an indication that higher frequencies should be adopted in this baseline-free challenging scenario.

With respect to the corrosion-like defects problem, ultrasonic signals due to the interaction of SH ultrasonic GW with corrosion-like defects in plates were applied to supervised ML algorithms for defect estimation. In the frequency-thickness product considered for this work, only either the SH0 or SH1 modes can propagate. They are thus mixed in time and frequency domain. Additionally, the individual behavior of each mode when interacting with the defect, which includes mode-conversion, is complex. These characteristics motivated the application of ML techniques. Moreover, we have investigated whether the estimation of the defect is facilitated and improved if one is able to remove the effect of mode-mixing. A baseline-free data-driven model was proposed and different damage indices for the defect, such as continuous and discrete, were studied. Both numerical and experimental signals were used, such that the former served to train the model and the latter to test it. We have shown that the mode separation strategy jointly with the SHM methodology proposed, has indeed significantly improved the performance of the data-driven models, since mode-mixing, mainly of converted modes, produced complicated signals. When full mode separation is applied, results such as the error rate improve by 30% and 67% in the regression-based and classification-based models, respectively.

Finally, it can be seen that different ML pipelines were developed in this work. These are the main contributions herein and can be reused, adapted, and generalized to others, whether being similar or different, engineering problems in the context of GW and ML.

9.1 Future Works

Based on the results and discussion presented at the full length of this thesis, upcoming works and activities can be suggested:

- Feature extraction methodologies based on a nonlinear auto-regressive moving average (NARMA) strategy should be considered for datasets that did not receive the baseline subtraction treatment in order to improve regularization of the parameters. Moreover, more research is needed in order to automate the process of obtaining the NARMA models, which are considerably more complex than the linear ones to build. Namely, one should define the model orders for the output and residuals and define the complexity control parameters in the forward regression orthogonal least-squares procedure for model estimation and term selection. This is needed as the results herein depicted indicate that it is possible to improve the classification accuracy by using nonlinear models.
- For the dot type defects, the disclosure of the position of the defect might be contemplated. The idea here is to train new models where the output would be the defect positions either in the x-axis and, also, in the y-axis. As the dataset in question is small, more data would be necessary. Therefore, data augmentation techniques should be investigated [17, 104].
- Another approach to be explored is deep learning, for example, recurrent neural networks over the raw data series. The latter would result in the

exemption of the feature extraction step and could possibly yield superior performance in the raw datasets as seen in recent works [105, 106].

- For the corrosion-type defects problem, one should compare the mode separation performance with more parameters, such as varying the position and length of the defect, which is already discussed in the literature with pipes and plates [107, 45].
- Another point to be discussed is the geometry of the defect, a more complex approach of defect generation can be further explored, creating defect geometries that are even closer to real corrosion defects.
- More robust models should be created for improving the performance of the current results of the corrosion-type defects problem. A data augmentation technique [108, 109, 110, 111, 112] could be implemented in order to increase the number of experimental signals and incorporate them in the training of more complex models such as convolutional neural networks [48, 17].

Bibliography

[1] MCCARTHY, J.. **What is artificial intelligence?**, 2007. Technical report, Computer Science Department, Stanford University.

[2] RUSSEL, S.; NORVIG, P.. **Artificial Intelligence: a Modern Approach**. Springer, Englewood, New Jersey, USA, 3rd edition, 2016.

[3] JERRY, M. M.. **Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions**. Springer, 2nd edition, 2019.

[4] HAYKIN, S.. **Neural Networks: a Comprehensive Foundation**. Prentice Hall PTR, Hamilton, Canada, 2nd edition, 1994.

[5] BACK, T.. **Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms**. Oxford university press, 1996.

[6] ALPAYDIN, E.. **Introduction to machine learning**. MIT press, 2020.

[7] QUIROZ, I. A.; ALFÉREZ, G. H.. **Image recognition of legacy blueberries in a chilean smart farm through deep learning**. Computers and Electronics in Agriculture, 168:105044, 2020.

[8] LEE, S.; KIM, H.; LIEU, Q. X. ; LEE, J.. **CNN-based image recognition for topology optimization**. Knowledge-Based Systems, p. 105887, 2020.

[9] KOUROU, K.; EXARCHOS, T. P.; EXARCHOS, K. P.; KARAMOUZIS, M. V. ; FOTIADIS, D. I.. **Machine learning applications in cancer prognosis and prediction**. Computational and Structural Biotechnology Journal, 13:8–17, 2015.

[10] ELAZIZ, M. A.; HOSNY, K. M.; SALAH, A.; DARWISH, M. M.; LU, S. ; SAHLOL, A. T.. **New machine learning method for image-based diagnosis of COVID-19**. PLoS One, 15(6):e0235187, 2020.

[11] NABIPOUR, M.; NAYYERI, P.; JABANI, H.; SHAHAB, S. ; MOSAVI, A.. **Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis**. IEEE Access, 8:150199–150212, 2020.

[12] LEE, T. K.; CHO, J. H.; KWON, D. S. ; SOHN, S. Y.. **Global stock market investment strategies based on financial network indicators using machine learning techniques.** Expert Systems with Applications, 117:228–242, 2019.

[13] RAMOS, S.; GEHRIG, S.; PINGGERA, P.; FRANKE, U. ; ROTHER, C.. **Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling.** In: 2017 IEEE INTELLIGENT VEHICLES SYMPOSIUM, p. 1025–1032. IEEE, 2017.

[14] CHAITANYA, G. K.; MARAGATHAM, G.. **Object and obstacle detection for self-driving cars using GoogLeNet and deep learning.** In: ARTIFICIAL INTELLIGENCE TECHNIQUES FOR ADVANCED COMPUTING APPLICATIONS, p. 315–322. Springer, 2020.

[15] JIMÉNEZ, L. A. E.; BENALCÁZAR, M. E. ; SOTOMAYOR, N.. **Gesture recognition and machine learning applied to sign language translation.** In: VII LATIN AMERICAN CONGRESS ON BIOMEDICAL ENGINEERING, p. 233–236. Springer, 2017.

[16] HUANG, F.. **Machine learning dialect identification**, Feb. 20 2018. US Patent 9,899,020.

[17] EWALD, V.; GROVES, R. M. ; BENEDICTUS, R.. **DeepSHM: a deep learning approach for structural health monitoring based on guided lamb wave technique.** In: SENSORS AND SMART STRUCTURES TECHNOLOGIES FOR CIVIL, MECHANICAL, AND AEROSPACE SYSTEMS 2019. International Society for Optics and Photonics, 2019.

[18] ABBAS, M.; SHAFIEE, M.. **Structural health monitoring (SHM) and determination of surface defects in large metallic structures using ultrasonic guided waves.** Sensors, 18(11):3958, 2018.

[19] MITRA, M.; GOPALAKRISHNAN, S.. **Guided wave based structural health monitoring: A review.** Smart Materials and Structures, 25(5):053001, 2016.

[20] FARRAR, C.; WORDEN, K.. **Structural Health Monitoring: A Machine Learning Perspective.** Wiley, 2013.

[21] GIURGIUTIU, V.. **Structural Health Monitoring: with Piezoelectric Wafer Active Sensors.** Elsevier, 2007.

[22] GIURGIUTIU, V.. **Lamb wave generation with piezoelectric wafer active sensors for structural health monitoring.** In: SMART STRUCTURES AND MATERIALS 2003: SMART STRUCTURES AND INTEGRATED SYSTEMS, p. 111–122. International Society for Optics and Photonics, 2003.

[23] MALLET, L.; LEE, B. C.; STASZEWSKI, W. J. ; SCARPA, F.. **Structural health monitoring using scanning laser vibrometry: II. Lamb waves for damage detection**. Smart Materials and Structures, 13(2):261–269, feb 2004.

[24] SU, Z.; YE, L. ; LU, Y.. **Guided Lamb waves for identification of damage in composite structures: A review**. Journal of Sound and Vibration, 295(3-5):753–780, 2006.

[25] BEN, B.; BEN, B.; VIKRAM, K. ; YANG, S.. **Damage identification in composite materials using ultrasonic based Lamb wave method**. Measurement, 46(2):904–912, 2013.

[26] SHAN, S.; CHENG, L.. **Mode-mixing-induced second harmonic A0 mode Lamb wave for local incipient damage inspection**. Smart Materials and Structures, 29(5):055020, 2020.

[27] CLOUGH, M.; FLEMING, M. ; DIXON, S.. **Circumferential guided wave EMAT system for pipeline screening using shear horizontal ultrasound**. NDT & E International, 86:20 – 27, 2017.

[28] WANG, S.; HUANG, S.; ZHAO, W. ; WEI, Z.. **3D modeling of circumferential SH guided waves in pipeline for axial cracking detection in ILI tools**. Ultrasonics, 56(Supplement C):325 – 331, 2015.

[29] DEMMA, A.; CAWLEY, P. ; LOWE, M.. **Scattering of the fundamental shear horizontal mode from steps and notches in plates**. The Journal of the Acoustical Society of America, 113(4):1880–1891, 2003.

[30] HIRAO, M.; OGI, H.. **An SH-wave EMAT technique for gas pipeline inspection**. NDT & E International, 32(3):127 – 132, 1999.

[31] ZHANG, H.; DU, Y.; TANG, J.; KANG, G. ; MIAO, H.. **Circumferential SH wave piezoelectric transducer system for monitoring corrosion-like defect in large-diameter pipes**. Sensors, 20(2):460, 2020.

[32] KUBRUSLY, A. C.; FREITAS, M. A.; VON DER WEID, J. P. ; DIXON, S.. **Interaction of SH guided waves with wall thinning**. NDT & E International, 101(May 2018):94–103, 2019.

[33] LEINOV, E.; LOWE, M. J. ; CAWLEY, P.. **Investigation of guided wave propagation in pipes fully and partially embedded in concrete**. The Journal of the Acoustical Society of America, 140(6):4528–4539, 2016.

[34] CAWLEY, P.; ALLEYNE, D.. **The use of Lamb waves for the long range inspection of large structures**. Ultrasonics, 34(2-5):287–290, 1996.

[35] SILVA, C.; ROCHA, B. ; SULEMAN, A.. **Guided Lamb waves based structural health monitoring through a PZT network system**. In: 2ND INTERNATIONAL SYMPOSIUM ON NDT IN AEROSPACE, 2010.

[36] NG, C.-T.. **On the selection of advanced signal processing techniques for guided wave damage identification using a statistical approach**. Engineering Structures, 67:50 – 60, 2014.

[37] KUHN, M.; JOHNSON, K.. **Applied Predictive Modeling**. Springer New York, 2013.

[38] SALEHI, H.; BURGUEÑO, R.. **Emerging artificial intelligence methods in structural engineering**. Engineering Structures, 171:170 – 189, 2018.

[39] LECUN, Y.; BENGIO, Y. ; HINTON, G.. **Deep learning**. Nature, 521(7553):436–444, 2015.

[40] DE FENZA, A.; SORRENTINO, A. ; VITIELLO, P.. **Application of artificial neural networks and probability ellipse methods for damage detection using Lamb waves**. Composite Structures, 133:390 – 403, 2015.

[41] DWORAKOWSKI, Z.; DRAGAN, K. ; STEPINSKI, T.. **Artificial neural network ensembles for fatigue damage detection in aircraft**. Journal of Intelligent Material Systems and Structures, 28(7):851–861, 2017.

[42] ALGURI, K. S.; MELVILLE, J. ; HARLEY, J. B.. **Baseline-free guided wave damage detection with surrogate data and dictionary learning**. The Journal of the Acoustical Society of America, 143(6):3807–3818, 2018.

[43] AHARON, M.; ELAD, M. ; BRUCKSTEIN, A.. **K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation.** IEEE Transactions on Signal Processing, 54(11):4311–4322, Nov 2006.

[44] DIB, G.; KARPENKO, O.; KORICHO, E.; KHOMENKO, A.; HAQ, M. ; UDPA, L.. **Ensembles of novelty detection classifiers for structural health monitoring using guided waves.** Smart Materials and Structures, 27(1):015003, nov 2017.

[45] GARCÍA-GÓMEZ, J.; GIL-PITA, R.; ROSA-ZURERA, M.; ROMERO-CAMACHO, A.; JIMÉNEZ-GARRIDO, J. A. ; GARCÍA-BENAVIDES, V.. **Smart sound processing for defect sizing in pipelines using EMAT actuator based multi-frequency Lamb waves.** Sensors, 18(3):802, 2018.

[46] YAACOUBI, S.; MOUNTASSIR, M. E.; FERRARI, M. ; DAHMENE, F.. **Measurement investigations in tubular structures health monitoring via ultrasonic guided waves: A case of study.** Measurement, 147:106800, 2019.

[47] SEN, D.; AGHAZADEH, A.; MOUSAVI, A.; NAGARAJAIAH, S.; BARANIUK, R. ; DABAK, A.. **Data-driven semi-supervised and supervised learning algorithms for health monitoring of pipes.** Mechanical Systems and Signal Processing, 131:524 – 537, 2019.

[48] MELVILLE, J.; ALGURI, K. S.; DEEMER, C. ; HARLEY, J. B.. **Structural damage detection using deep learning of ultrasonic guided waves.** AIP Conference Proceedings, 1949(1):230004, 2018.

[49] HESSER, D. F.; KOCUR, G. K. ; MARKERT, B.. **Active source localization in wave guides based on machine learning.** Ultrasonics, p. 106144, 2020.

[50] KARATAȘ, M. A.; GÖKKAYA, H.. **A review on machinability of carbon fiber reinforced polymer (CFRP) and glass fiber reinforced polymer (GFRP) composite materials.** Defence Technology, 14(4):318–326, 2018.

[51] SOUTIS, C.. **Fibre reinforced composites in aircraft construction.** Progress in Aerospace Sciences, 41(2):143–151, 2005.

[52] FIGUEIREDO, E.; PARK, G.; FIGUEIRAS, J.; FARRAR, C. ; WORDEN, K.. **Structural health monitoring algorithm comparisons using**

standard data sets. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2009.

[53] JIMÉNEZ, A. A.; MUÑOZ, C. Q. G. ; MÁRQUEZ, F. P. G.. **Dirt and mud detection and diagnosis on a wind turbine blade employing guided waves and supervised learning classifiers**. Reliability Engineering & System Safety, 184:2–12, 2019.

[54] JIMÉNEZ, A. A.; MÁRQUEZ, F. P. G.; MORALEDA, V. B. ; MUÑOZ, C. Q. G.. **Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis**. Renewable Energy, 132:1034–1048, 2019.

[55] MOLL, J.; KATHOL, J.; FRITZEN, C.-P.; MOIX-BONET, M.; RENNOCH, M.; KOERDT, M.; HERRMANN, A. S.; SAUSE, M. G. ; BACH, M.. **Open guided waves: online platform for ultrasonic guided wave measurements**. Structural Health Monitoring, 18(5-6):1903–1914, 2019.

[56] KUBRUSLY, A. C.; FREITAS, M. A.; VON DER WEID, J. P. ; DIXON, S.. **Mode selectivity of SH guided waves by dual excitation and reception applied to mode conversion analysis**. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 65(7):1239–1249, 2018.

[57] KUBRUSLY, A. C.; FREITAS, M. A.; VON DER WEID, J. P. ; DIXON, S.. **Dataset on reflection and transmission coefficients of ultrasonic shear horizontal guided waves in plates with wall thinning**. Data in Brief, 21:2179–2191, 2018.

[58] ROSE, J. L.. **Ultrasonic Guided waves in solid media**. Cambridge University Press, 2014.

[59] HUBER, A.. **Dispersion calculator user's manual**. German Aerospace Center (DLR), Augsburg, Germany (2019).

[60] LEE, B.; STASZEWSKI, W.. **Modelling of Lamb waves for damage detection in metallic structures: Part I. wave propagation**. Smart Materials and Structures, 12(5):804, 2003.

[61] LIU, H.; MOTODA, H.. **Feature extraction, construction and selection: A data mining perspective**, volumen 453. Springer Science & Business Media, 1998.

[62] MALETZKE, A. G.; FERRERO, C. A.; TIBES, C. M.; CHERMAN, E. A. ; ZALEWSKI, W.. **Medical time series classification using global and local feature extraction strategies**. Journal of Health Informatics, 9(3), 2017.

[63] LECUN, Y.; BENGIO, Y. ; OTHERS. **Convolutional networks for images, speech, and time series**. The Handbook of Brain Theory and Neural Networks, 3361(10):1995, 1995.

[64] HJALTASON, G. R.; SAMET, H.. **Properties of embedding methods for similarity searching in metric spaces**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5):530–549, 2003.

[65] GUYON, I.; GUNN, S.; NIKRAVESH, M. ; ZADEH, L. A.. **Feature Extraction: Foundations and Applications**. Springer, 2008.

[66] OPPENHEIM, A. V.; SCHAFER, R. W.. **Digital Signal Processing**. Prentice-Hall, 1975.

[67] SIDNEY BURRUS, C.; GOPINATH, R. A. ; GUO, H.. **Introduction to wavelets and wavelet transforms**. A Primer; Prentice Hall: Upper Saddle River, NJ, USA, 1998.

[68] VETTERLI, M.; KOVACEVIC, J.. **Wavelets and subband coding**. Número BOOK. Prentice-Hall, 1995.

[69] SÖDERSTRÖM, T.; STOICA, P.. **System Identification**. Prentice-Hall, Inc., 1988.

[70] YING, Y.; GARRETT, J. H.; OPPENHEIM, I. J.; SOIBELMAN, L.; HARLEY, J. B.; SHI, J. ; JIN, Y.. **Toward data-driven structural health monitoring: Application of machine learning and signal processing to damage detection**. Journal of Computing in Civil Engineering, 27(6):667–680, 2013.

[71] BILLINGS, S. A.. **Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains**. John Wiley & Sons, 2013.

[72] YING, Y.; HARLEY, J.; GARRETT, JR, J. H.; JIN, Y.; OPPENHEIM, I. J.; SHI, J. ; SOIBELMAN, L.. **Applications of machine learning in pipeline monitoring**. In: COMPUTING IN CIVIL ENGINEERING (2011), p. 242–249. 2011.

[73] BALANDA, K. P.; MACGILLIVRAY, H.. **Kurtosis: a critical review**. The American Statistician, 42(2):111–119, 1988.

[74] FERNÁNDEZ-DELGADO, M.; CERNADAS, E.; BARRO, S. ; AMORIM, D.. **Do we need hundreds of classifiers to solve real world classification problems?** The Journal of Machine Learning Research, 15(1):3133–3181, 2014.

[75] HASTIE, T.; TIBSHIRANI, R. ; FRIEDMAN, J.. **The Elements of Statistical Learning**. Springer Series in Statistics. Springer, New York, USA, 2nd edition, 2008.

[76] HOSMER JR, D. W.; LEMESHOW, S. ; STURDIVANT, R. X.. **Applied Logistic Regression**. John Wiley & Sons, 2013.

[77] BOTTOU, L.. **Stochastic gradient descent tricks**. In: NEURAL NETWORKS: TRICKS OF THE TRADE, p. 421–436. Springer, 2012.

[78] YPMA, T. J.. **Historical development of the Newton–Raphson method**. SIAM review, 37(4):531–551, 1995.

[79] SINGH, Y.; CHAUHAN, A. S.. **Neural networks in data mining**. Journal of Theoretical & Applied Information Technology, 5(1), 2009.

[80] RUMELHART, D. E.; HINTON, G. E. ; WILLIAMS, R. J.. **Learning representations by back-propagating errors**. Nature, 323(6088):533–536, 1986.

[81] ZHANG, Z.. **Improved Adam optimizer for deep neural networks**. In: 2018 IEEE/ACM 26TH INTERNATIONAL SYMPOSIUM ON QUALITY OF SERVICE, p. 1–2. IEEE, 2018.

[82] GOODFELLOW, I.; BENGIO, Y. ; COURVILLE, A.. **Deep Learning**. MIT Press, 2016.

[83] JANOCHA, K.; CZARNECKI, W. M.. **On loss functions for deep neural networks in classification**. arXiv preprint arXiv:1702.05659, 2017.

[84] HARRINGTON, P. D. B.. **Multiple versus single set validation of multivariate models to avoid mistakes**. Critical Reviews in Analytical Chemistry, 48(1):33–46, 2018.

[85] FEURER, M.; SPRINGENBERG, J. T. ; HUTTER, F.. **Initializing bayesian hyperparameter optimization via meta-learning**. In:

TWENTY-NINTH AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, 2015.

[86] HUTTER, F.; KOTTHOFF, L. ; VANSCHOREN, J.. **Automated machine learning: methods, systems, challenges**. Springer Nature, 2019.

[87] BERGSTRA, J.; BENGIO, Y.. **Random search for hyper-parameter optimization**. The Journal of Machine Learning Research, 13(1):281–305, 2012.

[88] METZ, C. E.. **Basic principles of ROC analysis**. In: SEMINARS IN NUCLEAR MEDICINE, número 4, p. 283–298. WB Saunders, 1978.

[89] CANTOR, A. B.. **Sample-size calculations for Cohen's kappa**. Psychological Methods, 1(2):150, 1996.

[90] DAFNA, E.; TARASIUK, A. ; ZIGEL, Y.. **Sleep-wake evaluation from whole-night non-contact audio recordings of breathing sounds**. PLoS one, 10(2):e0117382, 2015.

[91] DODSON, J.; INMAN, D.. **Thermal sensitivity of Lamb waves for structural health monitoring applications**. Ultrasonics, 53(3):677–685, 2013.

[92] CROXFORD, A. J.; MOLL, J.; WILCOX, P. D. ; MICHAELS, J. E.. **Efficient temperature compensation strategies for guided wave structural health monitoring**. Ultrasonics, 50(4-5):517–528, 2010.

[93] BACH, M.; POUILLY, A.; ECKSTEIN, B. ; BONET, M. M.. **Reference damages for verification of probability of detection with guided waves**. Structural Health Monitoring 2017, (SHM), 2017.

[94] DREHMER, D. E.; MORRIS, G. W.. **Cross-validation with small samples: An algorithm for computing Gollob's estimator**. Educational and Psychological Measurement, 41(1):195–200, 1981.

[95] MØLLER, M. F.. **A scaled conjugate gradient algorithm for fast supervised learning**. Neural Networks, 6(4):525–533, 1993.

[96] DIXON, S.; PETCHER, P. A.; FAN, Y.; MAISEY, D. ; NICKOLDS, P.. **Ultrasonic metal sheet thickness measurement without prior wave speed calibration**. Journal of Physics D: Applied Physics, 46(44):445502, 2013.

[97] PETCHER, P. A.; DIXON, S.. **Mode mixing in shear horizontal ultrasonic guided waves**. Nondestructive Testing and Evaluation, p. 1–20, 2016.

[98] ALLEYNE, D.; CAWLEY, P.. **A two-dimensional fourier transform method for the measurement of propagating multimode signals**. The Journal of the Acoustical Society of America, 89(3):1159–1168, 1991.

[99] WANG, S.; HUANG, S.; WANG, Q.; ZHANG, Y. ; ZHAO, W.. **Mode identification of broadband Lamb wave signal with squeezed wavelet transform**. Applied Acoustics, 125:91–101, 2017.

[100] ZHAI, G.-F.; LI, Y.-Q.. **Single SH guided wave mode generation method for ppm emats**. Chinese Physics B, 29(5):054303, 2020.

[101] KHALILI, P.; CEGLA, F.. **Excitation of single-mode shear-horizontal guided waves and evaluation of their sensitivity to very shallow crack-like defects**. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 2020.

[102] QIU, G.; SONG, X.; ZHANG, X.; TU, J. ; CHEN, T.. **Pure SH1 guided-wave generation method with dual periodic-permanent-magnet electromagnetic acoustic transducers for plates inspection**. Sensors, 19(13):3019, Jul 2019.

[103] KUBRUSLY, A. C.; VON DER WEID, J. P. ; DIXON, S.. **Experimental and numerical investigation of the interaction of the first four SH guided wave modes with symmetric and non-symmetric discontinuities in plates**. NDT & E International, 108:102175, 2019.

[104] KO, T.; PEDDINTI, V.; POVEY, D. ; KHUDANPUR, S.. **Audio augmentation for speech recognition**. In: SIXTEENTH ANNUAL CONFERENCE OF THE INTERNATIONAL SPEECH COMMUNICATION ASSOCIATION, 2015.

[105] KARIM, F.; MAJUMDAR, S.; DARABI, H. ; CHEN, S.. **LSTM fully convolutional networks for time series classification**. IEEE access, 6:1662–1669, 2017.

[106] SONG, X.; LIU, Y.; XUE, L.; WANG, J.; ZHANG, J.; WANG, J.; JIANG, L. ; CHENG, Z.. **Time-series well performance prediction based on long short-term memory (LSTM) neural network model**. Journal of Petroleum Science and Engineering, 186:106682, 2020.

[107] LIU, S.; DU, C.; MOU, J.; MARTUA, L.; ZHANG, J. ; LEWIS, F.. **Diagnosis of structural cracks using wavelet transform and neural networks**. NDT & E International, 54:9–18, 2013.

[108] LE GUENNEC, A.; MALINOWSKI, S. ; TAVENARD, R.. **Data augmentation for time series classification using convolutional neural networks**. In: ECML/PKDD WORKSHOP ON ADVANCED ANALYTICS AND LEARNING ON TEMPORAL DATA, 2016.

[109] LEMLEY, J.; BAZRAFKAN, S. ; CORCORAN, P.. **Smart augmentation learning an optimal data augmentation strategy**. IEEE Access, 5:5858–5869, 2017.

[110] RASHID, K. M.; LOUIS, J.. **Times-series data augmentation and deep learning for construction equipment activity recognition**. Advanced Engineering Informatics, 42:100944, 2019.

[111] HU, T.; TANG, T. ; CHEN, M.. **Data simulation by resampling—a practical data augmentation algorithm for periodical signal analysis-based fault diagnosis**. IEEE Access, 7:125133–125145, 2019.

[112] OH, C.; HAN, S. ; JEONG, J.. **Time-series data augmentation based on interpolation**. Procedia Computer Science, 175:64–71, 2020.

A
Scientific Contributions

The following scientific contributions were made during the development of this work:

1. **RIBEIRO, M.G.C.**; KUBRUSLY, A. C.; AYALA, H. V. H.; **Damage Detection in Composite Plates with Ultrasonic Guided-waves and Nonlinear System Identification.** In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI). Canberra, Australia: IEEE, 2020.

2. **RIBEIRO, M.G.C.**; KUBRUSLY, A. C.; AYALA, H. V. H.; DIXON, S. **Machine Learning Based Corrosion-like Defect Estimation with Shear-Horizontal Guided Waves Improved by Mode Separation** (manuscript submitted).