



**Christian Deyvi Villares Holguin**

**Mechanical Stress Estimation using Guided  
Ultrasonic Waves and Machine Learning**

**Dissertação de Mestrado**

Thesis presented to the Programa de Pós-graduação em Engenharia Elétrica, do Departamento de Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

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

Rio de Janeiro  
April 2022



**Christian Deyvi Villares Holguin**

**Mechanical Stress Estimation using Guided  
Ultrasonic Waves and Machine Learning**

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

**Prof. Alan Conci Kubrusly**

Advisor

Centro de Estudo em Telecomunicações – PUC-Rio

**Prof. Helon Vicente Hultmann Ayala**

Co–advisor

Departamento de Engenharia Mecânica – PUC-Rio

**Prof. Vivian Suzano Medeiros**

Centro de Pesquisa em Tecnologia de Inspeção – PUC-Rio

**Prof. Igor Braga de Paula**

Departamento de Engenharia Mecânica – PUC-Rio

Rio de Janeiro, April 29th, 2022

All rights reserved.

**Christian Deyvi Villares Holguin**

Electronic engineer graduated from Universidad Nacional San Antonio Abad del Cusco (Cusco, Peru) in 2018.

Bibliographic data

Villares Holguin, Christian Deyvi

Mechanical Stress Estimation using Guided Ultrasonic Waves and Machine Learning / Christian Deyvi Villares Holguin; advisor: Alan Conci Kubrusly; co-advisor: Helon Vicente Hultmann Ayala. – 2022.

69 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2022.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Ondas guiadas ultrasônicas. 3. Aprendizado automático. 4. Aprendizado supervisionado. 5. Aprendizado não-supervisionado. 6. estimação de tração. I. Kubrusly, Alan C.. II. Ayala, Helon V. H.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

## Acknowledgments

In my life there have always been people who have supported me, for which I will be eternally grateful and in debt with them.

First of all, I would like to thank my family, especially my parents Pedro and Edith, my two sisters Melissa and Pamela, my grandparents Victor, Maurica and Benedicta, my girlfriend Diana, who always motivate me and give me their support.

I would like to thank my advisor Prof. Dr. Alan Conci Kubrusly and my co-advisor Prof. Dr. Helon Vicente Hultmann Ayala, for giving me support, offering me the opportunity to work on several articles, being patient to help me understand the issues related to this work and helping me solve the most diverse problems that arose along the way.

Thanks to PUC-Rio for the master's scholarship provided. This study was funded in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Financial Code 001.

## Abstract

Villares Holguin, Christian Deyvi; Kubrusly, Alan C. (Advisor); Ayala, Helon V. H. (Co-Advisor). **Mechanical Stress Estimation using Guided Ultrasonic Waves and Machine Learning**. Rio de Janeiro, 2022. 69p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Due to the acoustoelastic effect, Ultrasonic Guided Waves (UGWs) have been used to estimate mechanical stress in a non-expensive and non-destructively fashion. Machine Learning (ML) has been applied to map complex waveforms to stress estimates, though important aspects, such as accuracy and hardware consumption, have not been explored. Previously in the literature, there are also not many works on the use of unsupervised learning for automatic labeling of samples with different stress states. Therefore, this thesis presents two approaches, (i) the supervised approach aims to propose a data modeling methodology that optimizes accuracy and computational implementation, for real-time ultrasonic based stress estimation and (ii) the unsupervised approach aims at comparing unsupervised frameworks to label a small dataset according to the stress state. For the former, shallow and deep learning models with dimensionality reduction were evaluated, these models are created and tested using a Monte-Carlo holdout procedure to evaluate their robustness under different stress conditions. The results show that, using shallow models and Principal Component Analysis (PCA), an accuracy improvement and hardware consumption as compared to the state of the art reported with deep neural network models were obtained. For the latter, dimensionality reduction methods: PCA and t-distributed stochastic neighbor embedding (t-SNE), are used to extract features from UGWs signals with different stress levels. The features are used to group the samples into low, medium and high stress states. A qualitative and quantitative analysis of the results was performed. Considering the analysis of metrics for clustering, PCA performed the best clustering, qualitatively, showing less overlapping of clusters than t-SNE. The two approaches used in this thesis, managed to extract meaningful features which helped in both estimation and stress labeling, contributing to the creation of more efficient ML models and in the problem of interpreting UGWs.

## Keywords

Guided Wave Ultrasonic; Machine learning; Supervised learning; Unsupervised learning; tensile stress estimation.

## Resumo

Villares Holguin, Christian Deyvi; Kubrusly, Alan C.; Ayala, Helon V. H.. **Estimação da tensão mecânica usando ondas ultrassônicas guiadas e machine learning**. Rio de Janeiro, 2022. 69p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Devido ao efeito acoustoelástico, as Ondas guiadas ultrassônicas (UGWs) têm sido usadas para estimar a tensão mecânica com baixo custo de forma não destrutiva. O Aprendizado de máquina (ML) tem sido aplicado para mapear formas complexas de ondas para estimar a tensão mecânica, embora aspectos importantes como precisão e consumo computacional não tenham sido explorados. Na literatura também não há muito trabalho sobre o uso do aprendizado não supervisionado para a rotulagem automática de amostras com diferentes estados de tensão. Portanto, esta tese apresenta duas abordagens: i) a abordagem supervisionada propõe uma metodologia de modelagem de dados que otimiza a precisão e a implementação computacional, para a estimação da tensão baseada em UGWs em tempo real e ii) a abordagem não supervisionada compara estruturas não supervisionadas para rotular um pequeno conjunto de dados de acordo com o estado de tensão. Para o primeiro, foram avaliados modelos de aprendizagem superficial e profunda com redução de dimensionalidade, estes modelos são criados e testados usando um procedimento de "hold-out" Monte-Carlo para avaliar sua robustez. Os resultados mostram que, utilizando modelos superficiais e Análise de componentes principais (PCA), foi obtida uma melhoria de precisão e no consumo de hardware em comparação com o estado da arte com modelos de redes neurais profundas. Para o segundo, métodos de redução de dimensionalidade: PCA e t-distributed stochastic neighbor embedding (t-SNE), são usados para extrair características de sinais UGWs. As características são usadas para agrupar as amostras em estados de baixa, média e alta tensão. Uma análise qualitativa e quantitativa dos resultados foi realizada, considerando a análise de métricas para agrupamento, o PCA realizou o melhor agrupamento, qualitativamente, mostrando menos sobreposição em grupos do que t-SNE. As duas abordagens utilizadas nesta tese, conseguiram extrair características significativas que ajudam tanto na estimativa quanto na rotulagem de dados, contribuindo para a criação de modelos de ML mais eficientes e no problema de interpretação de UGWs.

## Palavras-chave

Ondas guiadas ultrassônicas; Aprendizado automático; Aprendizado supervisionado; Aprendizado não-supervisionado; estimação de tração.

## Table of contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Objectives	16
1.2	Contributions	16
1.3	Organization	17
<b>2</b>	<b>Theoretical Background</b>	<b>18</b>
2.1	Ultrasonic Guided Waves (UGW)	18
2.2	Machine Learning	21
2.3	Supervised Learning	21
2.4	Unsupervised Learning	25
<b>3</b>	<b>Experimental Setup</b>	<b>31</b>
3.1	Measurement Campaign	32
<b>4</b>	<b>Improved Stress Estimation with Machine Learning and Ultrasonic Guided Waves</b>	<b>36</b>
4.1	Exploratory Analysis	36
4.2	Feature Extraction using PCA	38
4.3	Inputs for the Convolutional Neural Network	40
4.4	Resampling-Based Model Construction and Validation	41
4.5	Results	43
4.6	Discussion	49
<b>5</b>	<b>Unsupervised Tensile Stress Estimation Using Ultrasonic Guided Waves Signals in Plates</b>	<b>51</b>
5.1	Feature Extraction using t-SNE	51
5.2	Results	51
5.3	Discussion	56
<b>6</b>	<b>Conclusion and Future Works</b>	<b>57</b>
6.1	Future Works	58
<b>7</b>	<b>Bibliography</b>	<b>60</b>

## List of figures

Figure 2.1	Lamb waves, based on [44].	18
Figure 2.2	Dispersion curves, the phase velocity for a 3-mm-thick aluminum plate from 0 to 2.5 MHz.	20
Figure 2.3	Architectural model of ridge regression.	22
Figure 2.4	A schematic diagram of the support vector regression.	23
Figure 2.5	A flowchart of the k-means algorithm.	26
Figure 3.1	Experimental setup description. The lower plots illustrate the excitation signal (left) and the received signals (right).	31
Figure 3.2	Excitation signal (sinc like signal with 2.5 MHz bandwidth) and Spectrum.	32
Figure 3.3	Visual representation of all received signals with colors associated with the strain perceived on the plate. All received signals refer to the same time instant relative to the transmission.	33
Figure 3.4	Plot of four waveforms received under different stress conditions.	34
Figure 3.5	Histogram of the stress applied during all static experiments.	35
Figure 4.1	Cross-correlation matrix of all measurements made.	37
Figure 4.2	Cumulative explained variance of the principal components. Only 9 out of almost 500 components are sufficient to obtain 95%.	38
Figure 4.3	Principal components obtained from all measurements made (upper), ordered by ascending order of stress applied (lower).	39
Figure 4.4	Waveforms reshaped to images that can be fed to a CNN, by varying the magnitude of stress applied.	40
Figure 4.5	Monte-Carlo holdout approach using grid search and repeated 5-fold cross-validation.	42
Figure 4.6	Summary of the workflow for the construction and validation of the supervised models.	45
Figure 4.7	Distribution of the error for all holdout model predictions, in ascending order of MSE (upper to lower). It is interesting to note that the models perform differently depending on the stress condition.	46
Figure 4.8	Predictions made on all realizations for different holdout cases, organized by different model classes.	47
Figure 4.9	Model sizes, represented by mean and standard deviation, and respective MSE metrics in various holdout phases. Black dotted lines denote the region, in this space, that the two best models perform better with respect jointly to model performance and size.	48
Figure 5.1	Overview of the unsupervised stress estimation frameworks used for data labeling	52
Figure 5.2	K-means clusters represented in the space composed by first and second features derived from Principal Component Analysis (PCA), where ○ is high stress, × is medium stress and □ is low stress.	53

Figure 5.3	Half violin plots and boxplots for stress distribution representation of all k-means clusters derived from PCA. Purple, green and yellow colors represent cluster with low, medium and high stress states, respectively.	54
Figure 5.4	K-means clusters represented in the space composed by first and second features derived from t-SNE, where ○ is high stress, □ is medium stress and × is low stress.	55
Figure 5.5	Half violin plots and boxplots for stress distribution representation of all k-means clusters derived from t-SNE. Purple, green and yellow colors represent cluster with low, medium and high stress states, respectively.	56

## List of tables

Table 3.1	Transducer piezoelectric constructive characteristics	32
Table 4.1	Hyperparameter settings for the models tested in the present work using repeated k-fold cross-validation and randomized search.	44
Table 4.2	Squared error statistics of holdout realizations for all models tested, ordered in ascending order of MSE.	45
Table 4.3	Statistics for the model size in bytes for the models tested in the present work using repeated k-fold cross-validation and randomized search. Please note that LRG and CNN do not vary their sizes in each holdout resample.	48
Table 5.1	Clustering evaluation metrics.	53

## List of Abbreviations

AI – Artificial Intelligence

ANN – Artificial Neural Network

CH – Calinski-Harabasz Index

CNN – Convolutional Neural Network

CSET – Center for Security and Emerging Technology

DB – Davies-Bouldin Index

DTR – Decision Tree Regression

KL – Kullback-Leibler

KNN – k-Nearest Neighbors Regression

LRG – Ridge Regression

ML – Machine Learning

MSE – Mean Squared Error

NDT – Non-Destructive Evaluation

PCA – Principal Components Analysis

PC – Principal Components

PZT – Piezoelectric Transducer

SC – Silhouette Coefficient

SHM – Structural Health Monitoring

SVR – Support Vector Regression

SVD – Singular Value Decomposition

RNN – Recurrent Neural Networks

RFR – Random Forest Regression

t-SNE – t-Distributed Stochastic Neighbor Embedding

UGW – Ultrasonic Guided Wave

*Predicting the future isn't magic,  
it's artificial intelligence*

**Dave Waters, .**

# 1

## Introduction

The area of Structural Health Monitoring (SHM) is a multidisciplinary field that seeks to develop and implement methods to continuously monitor the health of structures in order to prevent failures and their associated costs, as well as to ensure safety [1]. Mechanical structures are constantly exposed to dynamic loads caused by vibration, thermal variations and other causes inducing fatigue failures [2, 3].

Stress estimation is particularly important in SHM applications [4], since its occurrence is directly related to fatigue, which can lead to catastrophic events when it exceeds safe levels. There are methods of evaluation of structures, and these are divided into semi-destructive and non-destructive. In the first case, which is not the focus of the present thesis, the most widely used is the hole drilling method [5–8] which demands space to allocate a drilling device and strain gauge rosettes, as well as presenting limitations regarding the thickness of the structure under study and the proximity of multiple measurements [9]. Non-destructive Testing (NDT) for stress monitoring has been performed recently. For instance, through i) the use of strain gauges [10, 11], but presenting limited resolution as a drawback; ii) digital image correlation [12, 13] which are not viable options for real-time continuous monitoring; iii) ultrasonic signals which can be suitable for stress estimation due to the sensitivity of the ultrasonic wave to the change of stress state of the medium [14–16].

Ultrasonic guided waves (UGWs) can be used to inspect wide areas, allowing a continuous monitoring of the structures in service. It has sensitivity to surface and internal defects, operating with inexpensive sensors in a wide frequency range according to the frequency-band characteristic of the load [17, 18]. The basic principle of using ultrasound to measure stress relies on observing subtle time-of-flight variations of a received pulse [19], since the propagation speed of ultrasonic waves is a function of the medium stress, according to the acoustoelastic theory [20, 21]. Ultrasonic guided waves propagate in several modes, each of which may be dispersive, meaning that their propagating speeds are frequency-dependent [22]. As opposed to bulk ultrasonic measurement, guided waves show more complex stress-dependence behavior since each mode presents a distinct sensitivity to stress, which also changes with frequency [23–27].

Therefore, using ultrasonic responses for stress estimation with broadband UGWs can be challenging by exploiting the direct underlying acoustoelastic

mechanism, since the received signal is altered in a nontrivial fashion being composed of subtle stress-induced changes in a complex waveform pattern. Nevertheless, they are fruitful to signal-processing techniques. For instance, the time-reversal process has been used before for strain monitoring [28]. It uses a signal response at null-stress as a reference for a time-reversed excitation, which produced a focused signal, whose quality deteriorates as the stress levels deviate from the reference.

Recently, the field of data-driven monitoring of mechanical systems, particularly with Machine Learning (ML) methods, has received a great deal of attention [29]. To mention a few recent research efforts, Vieira and Lambros [30] applied an Artificial Neural Network (ANN) to evaluate plastic strain using microstructures of metallic plates. Acoustic localization was performed in [31] using ANNs for structural health monitoring. Wu et al. [32] proposed the use of soft computing methods for enhancing digital image correlation methods for strain estimation. Several supervised ML approaches have been tested in [33] for fatigue prediction, showing overall better results for deep learning models. In [34] the authors evaluate the application of Convolutional Artificial Neural Networks (CNNs) for automatic feature extraction and stress estimation using elastic Lamb-wave measurements. In [35], wavelet Packet and Singular Value Decomposition were used to extract features from Ultrasonic Guided Wave (UGW) and fit a Support Vector Regressor model to estimate stress. In [36], a reduced-order spectral finite element model was used to generate UGW signals for the development of SHM systems based on deep recurrent and convolutional neural networks. The authors also compared deep techniques with conventional ML methods trained with handcrafted features, considering the performance of SHM systems in frequencies not used for training, as well as UGW signals corrupted with several Gaussian random noise levels to mimic experimental data.

Many engineering applications do not favor supervised learning, either because i) they do not allow collecting data from a class (e.g., task failure data Structural Health Monitoring (SHM)), generating imbalance problems; or ii) they are extremely expensive for data labeling, by presenting classes that are difficult to label or that do not represent a label very well [37]. These difficulties can be overcome by using unsupervised learning for data labeling.

In recent years, the use of small data approaches for training Machine Learning (ML) methods has been identified in [38], showing four approaches:

- **Transfer learning** is a ML method, first training the model in a data-rich environment, then “transferring” the features of the model to a task where it has little data.

- **Data Labeling** is an approach that starts with little labeled data, but plenty of unlabeled data. This approach uses a number of methods to make sense of the available unlabeled data, such as automatic generation of labels (automatic labeling).
- **Artificial data generation** this approach seeks to extract the maximum amount of information from a small amount of data by creating new data points or other related techniques.
- **Bayesian methods** that allow identifying which samples corroborate the most to reduce the uncertainties of predictions.

All of these approaches can facilitate working with large amounts of unlabeled data through automatic label generation. In the literature, there is a lack of works related to unsupervised clustering methods for labeling mechanical stress data, but there are some works related to clustering algorithms for damage detection. For example, the use of k-medoids and DBSCAN algorithms to detect and locate damage using the discrepancy matrix [39], the use of k-means algorithm with emission acoustic signals for damage detection [40].

It can thus conclude that, the recent literature review reveals that stress estimation based on ML and UGW techniques is becoming an attractive area for researchers in the SHM field. Nevertheless, there are gaps in the literature investigation, either in supervised or unsupervised learning, concerning the following specific points:

- (i) Supervised learning for stress estimation has not been fully explored towards devising a proper data workflow for building robust and computationally efficient predictive models. i) The work [34] evaluated solely the application of CNNs for the task, which was important for proving the concept, but failed short to devise best practices, evaluate the model using cross-validation to infer the model robustness, and also did not perform any optimization of model size in any form. As stress estimation should ideally be performed online and many times in a decentralized fashion, the model construction process should also take this point into account if the models are to be deployed using scarce, expensive, and energy-limited embedded computational resources.
- (ii) The role played by big data sets in the rise of Artificial Intelligence (IA) in the last decade is undeniable, but small data approaches is an area that has grown a lot recently, as reported by the Center for Georgetown University's Center for Security and Emerging Technology (CSET) [38]. These approaches help in situations where little or no labeled data is

available. Transfer learning is being used more for the purpose of data labeling, but no research was found using unsupervised approaches for data labeling using UGWs signals for stress estimation.

This thesis aims to contribute to the aforementioned points.

## 1.1 Objectives

The main objectives of this thesis are: i) to develop and improve the use of ML algorithms; ii) to constitute a framework for real-time ultrasonic-based stress estimation, feature extraction process and training of ML models through a supervised and unsupervised approach; and iii) to demonstrate the potential of ML methods to assist for real-time ultrasonic-based stress estimation and data labeling for UGWs in different values of stress.

To achieve the overall objectives of the thesis, the following steps were performed: i) to propose a data modeling methodology that optimizes accuracy and computational implementation, in order to devise best practices for real-time ultrasound-based stress estimation; ii) propose the use of clustering algorithms for data labeling; iii) to produce quantitatively and qualitatively valid results for supervised and unsupervised models.

## 1.2 Contributions

In the supervised approach, the following contributions are made:

- (i) A conceptually different data ingestion workflow than [34], with the use of Principal Component Analysis (PCA) to extract the characteristics of the guided wave signals, feeding shallow models.
- (ii) It is used Monte-Carlo holdout with a repeated cross-validation model validation procedure for supervised model evaluation and comparison.
- (iii) All results were obtained taking into account model size, hardware consumption and prediction error, which are important for the selection of the ideal model.

In the unsupervised approach, the following contributions are made:

- (i) Conventional dimensional reduction methods such as PCA and t-Distributed Stochastic Neighbor Embedding (t-SNE) are used to extract the characteristics of the guided wave signals, feeding a k-means clustering activity.

- (ii) Comparison of how conventional dimensional reduction techniques affect the stress clustering task. A qualitative comparison was made with the results obtained by each method.
- (iii) All the frameworks used for the unsupervised labeling of stress states are made considering small experimental data sets.

As a result of the aforementioned contributions to the field of SHM and ML, two articles were written during the master's period:

- One paper accepted in *Experimental Mechanics*, entitled “*Improved Stress Estimation with Machine Learning and Ultrasonic Guided Waves*”, *vol 62, pp 237–251 (2022)*, that it is located in appendix.
- One paper to be submitted, entitled “*Deep Unsupervised Tensile Stress Estimation Using Ultrasonic Guided Waves Signals in Plates*”.

### 1.3 Organization

The rest of the thesis is structured as follows: Chapter 2 introduces the general concepts necessary for understanding the data source and the description of the automatic learning models; Chapter 3 presents the description of the experimental setup; Chapter 4 presents the construction and validation of shallow models for estimation of stress, data ingestion workflow with exploratory analysis of data and feature extraction using PCA. It is explained the validation and the construction of the models, which use a Monte-Carlo holdout procedure with a k-fold repeated random hyperparameter. Resulting plots for all procedures are also shown, followed by a brief discussion of the obtained results; Chapter 5 details, the features extraction methods and the clustering methods that were used for estimation of stress. Resulting plots for all procedures are also shown, followed by a brief discussion of the obtained results. Finally, in chapter 6, the research is summarized, presenting the conclusions and giving guidelines for further implementations in order to improve the proposed method.

## 2 Theoretical Background

This chapter is dedicated to introduce basic concepts and describe the methods used in this thesis. This chapter is organized as follows: Section 2.1 introduces the basic concepts of ultrasonic guided waves; Section 2.2 introduces the concept of Machine Learning; Section 2.3 presents the definition of Supervised Learning and describes some supervised models; Section 2.4 presents the definition of unsupervised learning and describes some unsupervised models.

### 2.1 Ultrasonic Guided Waves (UGW)

UGW are ultrasonic waves that propagate through bonded structural media. One of the most common types of guided waves is the Lamb wave. This kind of guided wave is widely used in structural health monitoring and damage detection, because they can propagate long distances in the plate [41–43].

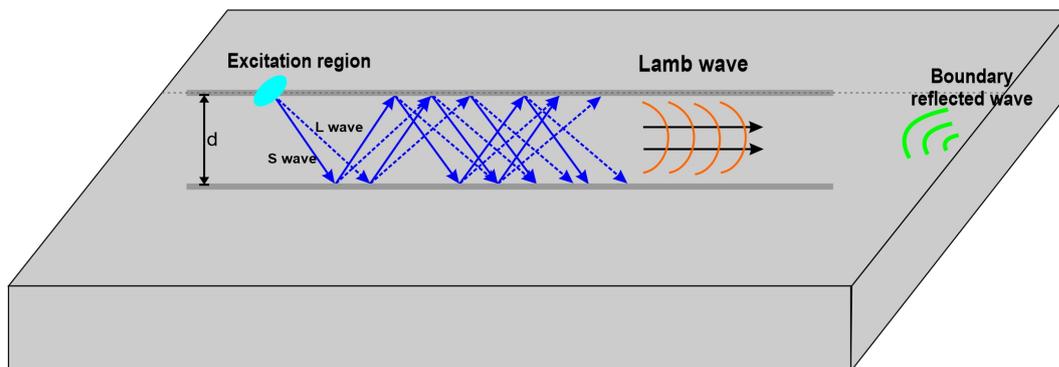


Figure 2.1: Lamb waves, based on [44].

Lamb waves are composed of several reflected waves, propagating along a homogeneous plate, as schematically shown in Fig. 2.1. Also, a Lamb wave propagates in symmetric or antisymmetric modes. The rise of the modes depend on frequency, i.e. as frequency increases, more modes of transportation arise, and there are countless numbers of them [28]. All of them are dispersive. Which means that the speed of propagation depends on the frequency [22]. Lamb wave motion has asymptotic behavior at low frequency and at high frequency [45]. At low frequency, the symmetric mode resembles axial waves, while the antisymmetric mode resembles bending waves. At high frequency, both the fundamental symmetric and antisymmetric waves tend to the Rayleigh waves,

as the particle motion is strong at surfaces and decays rapidly in thickness, show no dispersion and the same wave speed as the Rayleigh wave [22].

The propagation of each guided wave mode is dictated by its dispersion curves. The dispersion curve is the locus of pairs wavenumber and frequency, or phase speed and frequency in which propagation of a given wave mode occurs. It describes the propagation characteristics of Lamb waves and the natural resonance of a material [22, 45]. In order to obtain the dispersion curve for Lamb waves, one has to solve the Rayleigh-Lamb equations [22], either for a symmetric Eq. 2-1 or antisymmetric Eq. 2-2 mode that propagate in a plate of thickness  $2h$ , given respectively by:

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{4k^2pq}{(q^2 - k^2)^2} \quad (2-1)$$

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{(q^2 - k^2)^2}{4k^2pq} \quad (2-2)$$

the parameters  $p$  and  $q$  are defined by  $p^2 = \frac{\omega^2}{(c_L)^2} - k^2$  and  $q^2 = \frac{\omega^2}{(c_T)^2} - k^2$ , where  $\omega$  is angular frequency,  $k$  is wavenumber,  $c_L$  is the bulk longitudinal velocity and  $c_T$  is the bulk shear velocity. There is no analytical solution for Eqs. 2-2 and 2-1 which require a numerical solution. The group speed is defined as

$$c_g = \frac{d\omega}{dk} \quad (2-3)$$

where  $k$  is the wavenumber and  $\omega$  is the angular frequency. The phase speed is defined as

$$c_p = \frac{\omega}{k} \quad (2-4)$$

In Fig. 2.2 it is shown the lamb wave dispersion curve, presented as a plot of the phase speed versus the frequency-thickness product of an aluminum plate for frequencies up to 2.5 MHz. Also, it shows the kinds of waves that could propagate in the plate, i.e., there are symmetric (line blue) and antisymmetric (line red) modes that propagate in the plate, the firsts of each mode are denominated fundamentals S0 and A0, respectively, and they propagate at any frequency, as show in the Fig. 2.2. In addition, there are more symmetric modes, such as S1, S2, S3, and antisymmetric modes as A1, A2, A3, which are named according to their appearance.

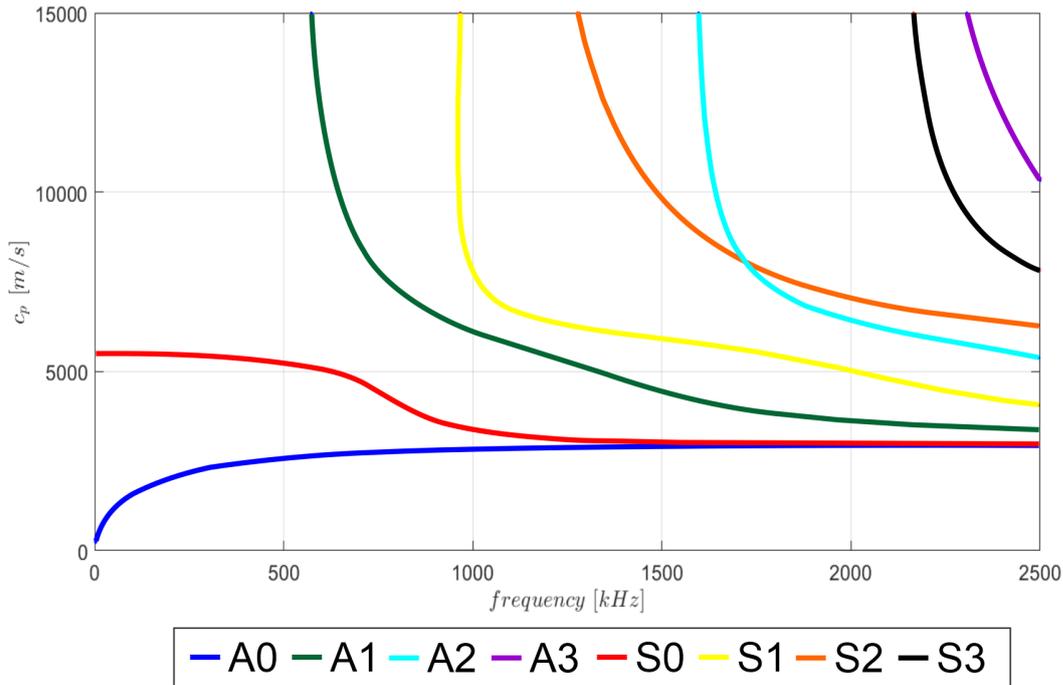


Figure 2.2: Dispersion curves, the phase velocity for a 3-mm-thick aluminum plate from 0 to 2.5 MHz.

### 2.1.1

#### Acoustoelastic effect in guided waves

Acoustoelasticity is the theory that rules the dependence of the propagation related phenomena of acoustic waves with elastic deformation in a solid, and an interesting phenomenon of such interaction is the change of wave velocities due to initial stresses or initial deformations in a preformed body, generally known as acoustoelastic effect [46].

The prior knowledge of the acoustoelastic behavior of the material under the stress state that it may be subject to is important, so is required to draw attention to excitation of guided waves and their modes. Usually, one works at low-frequency, before the cut-off frequency of higher-order modes, so there are fewer propagating modes [34] or with a single guided wave mode [47, 48]. Due to this phenomenon, time-of-flight change of waves is caused, which makes the measurement of stress with ultrasonic waves possible.

When several guided wave modes propagate simultaneously, which happens due to a broadband excitation, for instance, the received signal is composed by the superposition of the modes' signal which, depending on several factors, such as, the receiver position and excitation signals, can mix in the received signal [28, 49]. A broadband signal generates modes that present different stress-induced time shifts, including modes that may be even virtually non-sensitive to stress, which does not contribute to stress monitoring. However,

other modes may present considerable stress sensitivity, that somehow compensate for it. For instance, it has been observed that higher-order modes, close to their cut-off frequency, show a very intense speed variation [24].

The received signal becomes even more complex in the presence of features in the structure, such as defect, end reflection, or thickness variation, which produces scattering or mode-conversion.

## 2.2 Machine Learning

ML is generally considered to be a subfield of artificial intelligence [50]. It is defined as the use of various algorithms to teach computers to identify patterns in data which can be used for predicting future scenarios. ML provides computers with the ability to learn without being explicitly programmed [51]. ML can be divided into 4 types [52]: i) supervised learning; ii) unsupervised learning; iii) semi-supervised learning; iv) reinforced learning, the first two of which will be explained in the following sections. Before going into the definition of supervised learning, a data-driven model is defined, as follows:

$$\hat{y} = f(X; \theta), \quad (2-5)$$

where  $\hat{y} \in \mathbb{R}^n$  is the estimated value of the output  $y$ ,  $X \in \mathbb{R}^{n \times p}$  is the independent variable or the vector of inputs for the model  $f(\cdot) : \mathbb{R}^n \times \mathbb{R}^p \mapsto \mathbb{R}^n$ , and  $\theta$  is the vector that contains all the parameters that are obtained throughout the learning procedure. The structure of the model  $f(\cdot)$  depends on its class, as explained in Sec. 2.3.

The error may be then defined as

$$e = y - \hat{y}, \quad (2-6)$$

and is frequently used in the training stage to learn the parameters with optimization methods based on metrics such as the mean squared error:

$$\text{MSE} = \frac{1}{|T|} \sum_{i=1}^{|T|} e_i^2, \quad (2-7)$$

where  $T$  is the number the data points.

In the following, it is discussed the learning problem, together with the models used for evaluating the model construction procedure.

## 2.3 Supervised Learning

The general goal of supervised machine learning is to build  $f(\cdot)$ , that is, to define its structure and free-parameters  $\theta$ , based on measurements made on

$(X, y)$  such that it can generalize well and predict  $\hat{y}$  on unseen data [53]. In the following, it is presented the supervised methods used in this thesis:

- **Ridge Regression (LGR)**

Ridge regression shrinks the regression coefficients by imposing a penalty on their size [54]. In other words, the model is represented by

$$y_i = x_{i,*}\beta + \epsilon_i = \beta_1 x_{i,1} + \dots + \beta_p x_{i,p} + \epsilon_i \quad (2-8)$$

where  $\beta = (\beta_1, \dots, \beta_p)^T$  is the regression parameter, and Fig. 2.3 shows the architecture of the model based on [55].

The ridge regression estimator minimizes the ridge loss function [54], which is defined as:

$$\hat{\beta}^{ridge} = \arg \min_{\beta} \left( \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right) \quad (2-9)$$

this loss function is the traditional sum of squares augmented with ridge penalty defined by:  $\lambda \sum_{j=1}^p \beta_j^2$  where  $\lambda$  is the penalty parameter.

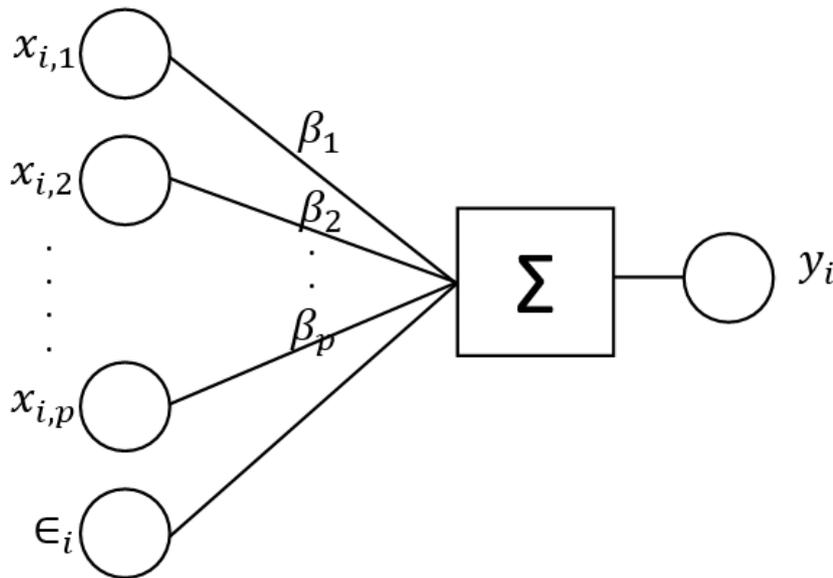


Figure 2.3: Architectural model of ridge regression.

It is maybe the simpler predictive model there is, as the output is solely the linear combination of the inputs, and works well under the assumption that the patterns are linearly separable [54].

- **Support Vector Regression (SVR)**

Among the most robust and precise machine learning methods [56], the SVR has an efficient training procedure. The goal of support vector-based training is to create hyperplanes that are placed in the input

space to optimize the margins for pattern separation [57], using the one-dimensional example in Fig. 2.4 based on [58]. The SVR model construction problem is defined as

$$\min_{w, \xi, \xi^*} \left[ \frac{1}{2} w^\top w + C \sum_i \xi_i + \xi_i^* \right] \quad (2-10a)$$

$$\text{subject to} \quad \begin{cases} \hat{y}_i - y_i \leq \epsilon + \xi_i, \\ y_i - \hat{y}_i \leq \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, i = 1, \dots, n, \end{cases} \quad (2-10b)$$

where  $\hat{y} = f(x_i, \theta) = w^\top \phi(x_i) + b$  is the model prediction,  $w, b \in \theta$  are the model parameters,  $C$  is the regularization factor,  $\xi, \xi^*$  ensures that it penalizes the samples that are  $\epsilon$  farther away from  $y$ , where the optimal hyperplane has a size  $2\epsilon$  and  $\phi(x_i) = \gamma K(x_i - x'_i)$  is the kernel function that maps the input vector which is typically defined as radial basis functions, higher-order polynomials, sigmoid, or linear functions, and they may also be customized [58].

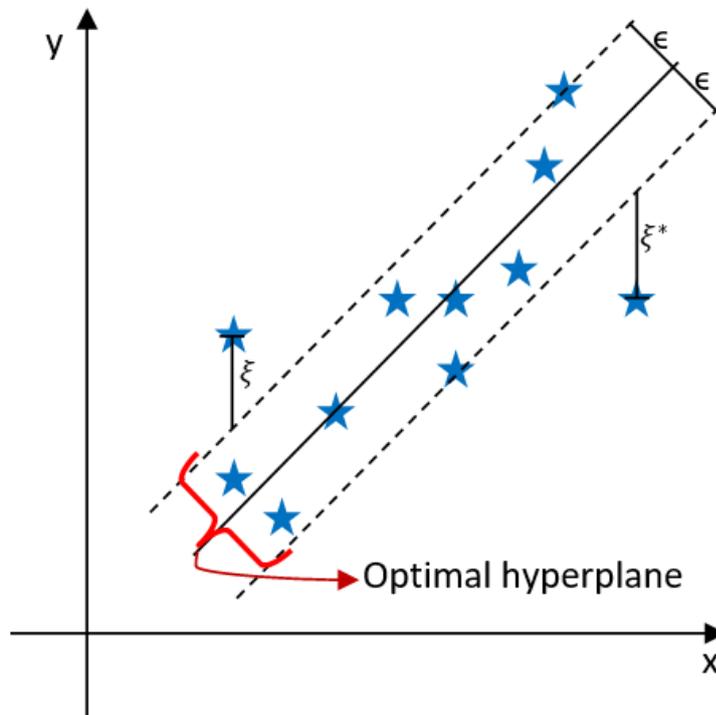


Figure 2.4: A schematic diagram of the support vector regression.

The SVR model estimation problem in (2-10) may be solved more conveniently by its dual formulation, for details see e.g. [59] or more recent developments in [60].

- **Decision Tree Regression (DTR)**

A DTR model is composed of nodes and leaves and may be represented as [61]

$$\hat{y} = f(x; \theta) = \sum_k c_k 1_{(x \in R_k)} \quad (2-11)$$

where  $c_k$  is the  $k$ -th regression target coefficient obtained for each non-overlapping region  $R_k$ , which are high-dimensional rectangles of the input space. Both  $c_k, R_k$  are defined through an iterative process that aims at maximizing the success of partitions used in the training samples on each node of the tree, such as the MSE or Friedman MSE. The size of the tree is controlled by its maximum depth, and also with the number of features in the input space used in each leaf for splitting.

- **Random Forest Regression (RFR)**

The random forest models have been proposed by Breiman [62]. The RFR is an ensemble of individual simpler DTR models by using bootstrap, which consists of random sampling data with replacement, and randomly selected features.

The RFR ensemble model is created as the following. Firstly, for each bootstrap drawn from the original dataset, it randomly selects a subset of the total features. Thus, each bootstrap is used to create a specialized tree for a random subset of the input space. The final prediction is output by simple average aggregation of the individual trees. Introductory material about tree-based models with computational code can be found in [61].

- **k-Nearest neighbors Regression (KNN)**

The KNN model is an instance-based inference mechanism, meaning that it does not need any training process for model construction. It simply stores all training data and during the inference stage, it compares the input to that of the  $k$ -nearest neighbors. The output of the model is the simple sum weighted of the output values for each of the  $k$ -nearest neighbors.

The only hyperparameters of the model are the number of  $k$  neighbors to consider when calculating the output and the neighboring metric. The KNN may be costly to run, since for each inference a norm metric should be calculated for the whole training set, which in many cases may render its use unfeasible. For details regarding KNN modeling, see e.g. [63] and a recent review in [64].

- **Convolutional Neural Networks (CNN)**

A fully connected Artificial Neural Network (ANN), summarily speaking, may be represented by a nested function composition as

$$\hat{y} = g_1 \circ \dots \circ g_N(X) \quad (2-12)$$

where  $g_i(\cdot)$  is the  $i$ -th activation function of the  $N$  layers calculated using learned parameters such as weights and biases. A deep convolutional artificial neural Convolutional Neural Network (CNN) is a special class of artificial neural network Artificial Neural Network (ANN), where typically the first layers of the network in (2-12) are constructed using convolution operators and many hidden layers. The convolution filter masks, as well as each of the weights and biases of such layers, are also learned throughout the training stage. This is very convenient since it may be interpreted as a joint feature and model joint construction, which made the application of CNN to highly complex and data-hungry problems such as image processing very common. The parameters are usually defined using stochastic gradient descent-based algorithms, such as Adam [65], employed in the present thesis. A gentle introduction on the topic of deep learning is given in [66], while recent reviews on deep learning are stated in [67, 68], and a position paper by its creators can be found at [69].

## 2.4

### Unsupervised Learning

Unsupervised learning is a type of ML, in which the main goal is to discover hidden and interesting patterns in the unlabeled data. The difference from supervised learning is that one does not need the output values to compare a prediction [52]. In the following, it is presented the unsupervised methods used in this thesis:

- **Clustering**

Clustering is the process that involves automatically discovering the natural grouping of input data, which can be a valuable information if you are looking for underlying meaning [70].

- **K-means:**

The k-means is an unsupervised learning algorithm used for clustering data, by finding groups with similar characteristics. The k-means algorithm can be summarized in the following steps [70]:

- (i) Choose “k” centroids, where “k” is the number of clusters provided.
- (ii) Selects “k” centroids at random locations among the data.
- (iii) Calculate the Euclidean distance between each point and the centroid. Each point is then assigned to the nearest cluster, taking the minimum distance.
- (iv) Isolate the clustered points along with their respective centroid, calculate the mean data point in the cluster and move the previous centroid to the location of the mean.
- (v) Repeat until convergence or until the maximum iteration limit has been reached.

In Fig. 2.5, a flowchart of the k-means algorithm is shown.

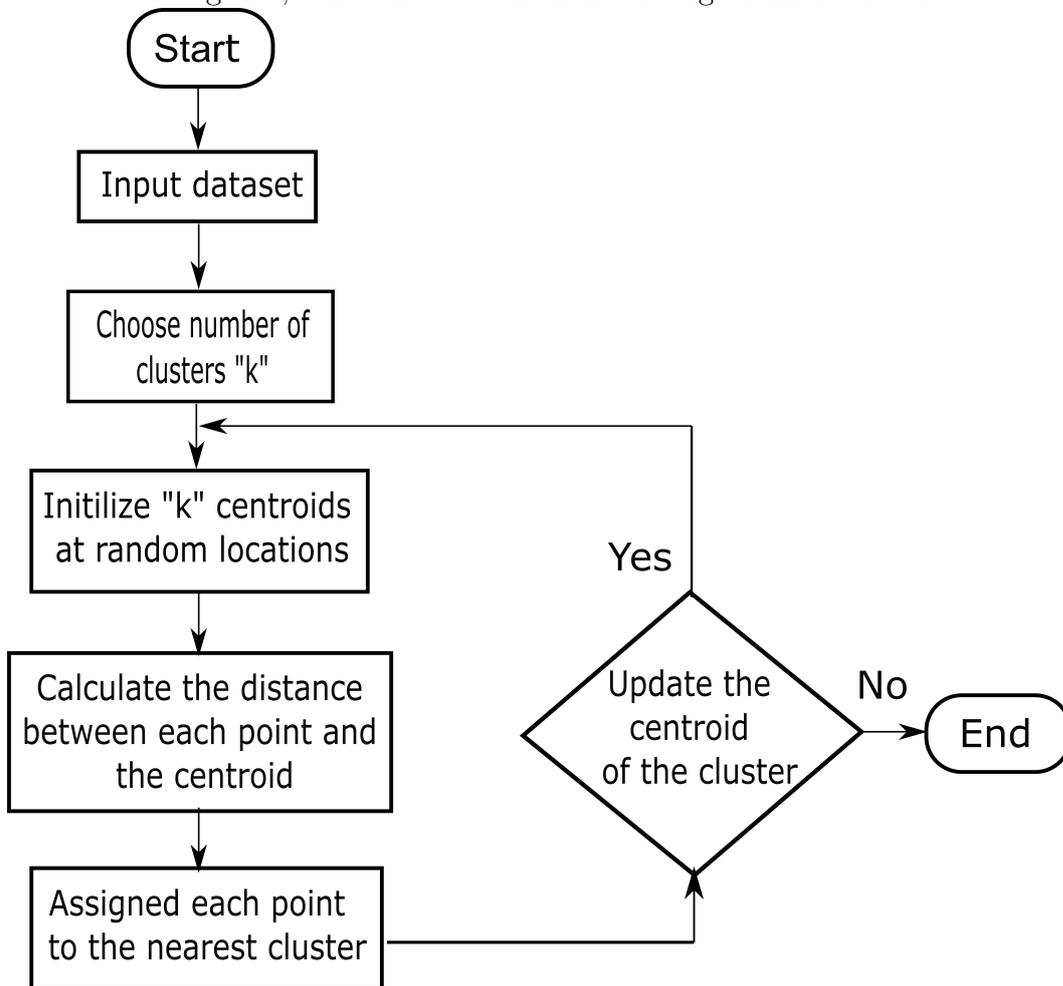


Figure 2.5: A flowchart of the k-means algorithm.

### • Clustering Performance

- \* **Silhouette Coefficient (SC)** The Silhouette Coefficient (SC) or silhouette score is a metric that analyzes how well a point fits into a cluster. SC is defined by [50, 70]:

$$SC = \frac{b - a}{\max(a, b)} \quad (2-13)$$

where  $a$  is the mean intra-cluster distance, i.e., is the mean distance between each point within a cluster,  $b$  is the mean inter-cluster distance, i.e., the mean of the nearest cluster distance for every sample. SC ranges from -1 to +1. A coefficient close to +1 means that the clusters are well separated from each other and are clearly distinguishable, while a coefficient close to 0 means that it is close to the cluster boundary, and finally a coefficient close to -1 means that clusters are assigned in the wrong way [50].

- \* **Calinski-Harabasz Index (CH)** Calinski-Harabasz Index (CH) evaluates the cluster validity based on the sum of the inter-cluster and the intra-cluster dispersion for all clusters, where dispersion, in this case, is the sum of the square of distances [71, 72]. For a dataset  $E$  of size  $n$ , clustered in  $k$  clusters, CH is defined:

$$CH = \frac{(n - k) \operatorname{tr}(B_k)}{(k - 1) \operatorname{tr}(W_k)} \quad (2-14)$$

where  $\operatorname{tr}(B_k)$  is a trace of the inter-cluster dispersion matrix and  $\operatorname{tr}(W_k)$  is a trace of the intra-cluster dispersion matrix defined by:

$$W_k = \sum_{q=1}^k \sum_{x \in D_q} (\|x - c_q\|^2) \quad (2-15a)$$

$$B_k = \sum_{q=1}^k n_q (\|c_q - c_E\|^2) \quad (2-15b)$$

with  $D_q$  the set of points in the cluster  $q$ ,  $c_q$  is the center of the cluster  $q$ ,  $c_E$  the center of  $E$ , and  $n_q$  the number of points in the cluster  $q$ . A higher value of the CH index means that the clusters are dense and well separated, although there is no “acceptable” cut-off value.

- \* **Davies-Bouldin Index (DB)** The goal of Davies-Bouldin Index (DB) is to maximize the distance between clusters and at the same time tries to minimize the distance between points in a cluster [73, 74].

The DB is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} R_{ij} \quad (2-16)$$

where  $R_{ij}$  is a measure of how good the clustering scheme is defined by:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \quad (2-17)$$

where  $s_i$  is the average distance between each point of cluster  $i$  and the centroid of that cluster,  $d_{ij}$  is the distance between cluster centroid  $i$  and  $j$ , and  $k$  is the number of clusters. The closer to zero its value, the better the partition. The computation of DB is simpler than that of SC.

### • Dimensionality Reduction

Dimensionality reduction is an unsupervised learning technique, which produces a more compact and easily interpretable representation of the data. This process helps to simplify the modeling of complex problems, eliminate redundancy and reduce the possibility that the model will overfit and therefore include results that do not match. One application of this technique is feature selection and feature extraction [70]. In the following, two methods of dimensionality reduction used in this thesis are introduced:

- **Principal Components Analysis (PCA)** The PCA can be used to perform dataset dimensionality reduction. The vast applications of PCA in many different areas are reviewed in [75], together with a deeper analysis of the matrix operations. Consider the following singular value matrix decomposition (SVD) [76]

$$X = U\Sigma V^T \quad (2-18)$$

where  $U \in \mathbb{R}^{n \times n}$ ,  $V \in \mathbb{R}^{p \times p}$  have orthonormal columns and, for the case that  $n < p$ ,

$$\Sigma = \left[ \text{diag}(\varsigma_1, \dots, \varsigma_p) \quad 0 \right], \quad (2-19)$$

is a matrix with diagonal elements in descending order of magnitude and zero elsewhere.

The  $m$  principal components (PC) of the matrix  $X$  may be retrieved by

$$X_R = [V_{:m}^T X^T]^T \quad (2-20)$$

where  $V_{:m}$  are the  $m$  first columns of  $V$  and  $X_R \in \mathbb{R}^{n \times m}$  represents the reduced dimension matrix containing the principal components. As the coefficients of  $\Sigma$  are ordered and  $U, V$  have orthonormal columns, the SVD may be interpreted as a matrix approximation method. For defining a hard threshold for  $m$ , one may measure

the approximation effectiveness of a set of the most informative principal components by the amplitude of  $\varsigma_i$ . As in [75], let

$$S = \sum_{i=1}^n \varsigma_i^2 \quad (2-21)$$

represent the cumulative sum of all components in the diagonal of  $\Sigma$  and

$$S_m = \sum_{i=1}^m \varsigma_i^2 \quad (2-22)$$

denote the cumulative sum of the  $m$  components that have higher amplitude in  $\Sigma$ . One can measure

$$E_m = \frac{S_m}{S} \cdot 100\%, \quad (2-23)$$

which is referred to as explained variance, representing the ratio of the cumulative sum of the squared values of the variances. Being so, one may set a threshold of e.g.  $E_m > 95\%$  in the data so that this amount of variance is present in the matrix approximation.

PCA provides interesting results for modeling, particularly when  $m \ll p$ . This means that much fewer dimensions are needed to approximate the original  $X$ , as the data-driven modeling procedure greatly benefits of such summarizing procedure in many cases by avoiding dealing with redundant information during the training procedure.

- **t-distributed stochastic neighbor embedding (t-SNE)**

t-SNE was developed in 2008 [77] and is an unsupervised nonlinear dimensionality reduction technique that is commonly used in data exploration and visualization of high dimensional data.

Consider the input data set (high dimensional) is

$$D = (x_1, x_2, \dots, x_N) \quad (2-24)$$

and a function

$$d(x_i, x_j) = \|x_i - x_j\| \quad (2-25)$$

that computes a Euclidean distance between a pair of points. The t-SNE constructs two probability distributions, the first is a Gaussian distribution over the pairs of points in the high dimensional space [78, 79]:

$$p_{i|j} = \frac{\exp\left(-\frac{d(x_i, x_j)^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d(x_i, x_k)^2}{2\sigma_i^2}\right)} \quad (2-26)$$

where  $\sigma_i$  is the variance of the Gaussian that is centered on the data point  $x_i$ . The conditional probabilities are symmetrized by averaging the two probabilities:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \quad (2-27)$$

this is done in the high dimensional space to obtain final similarities in this space. Each point in the high-dimensional space is mapped to a low-dimensional space, obtaining a set that is

$$E = (y_1, y_2, \dots, y_N) \quad (2-28)$$

with  $y_i \in \mathbb{R}^s$  (typical values for  $s$  are 2 or 3). The mapping is based on the similarity pairwise of the points in the high-dimensional space. Where it is obtained a set (low dimensional), that is,  $E = (y_1, y_2, \dots, y_N)$  with  $y_i \in \mathbb{R}^s$  (typical values for  $s$  are 2 or 3). Having low-dimensional counterparts  $y_i$  and  $y_j$  of the high-dimensional data points  $x_i$  and  $y_j$ , it calculates the joint probability over the pairs of points in the low-dimensional space, using heavy-tailed Student-t distribution [78, 79] with one degree of freedom:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}. \quad (2-29)$$

To find the low-dimensional data representation that minimizes the mismatch between  $P$  and the joint probability distribution  $Q$  based on Student-t, a gradient descent method based on Kullback-Leibler divergence (KL) is used:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (2-30)$$

One can enumerate the following advantages of t-SNE.

- It is a non-linear dimensionality reduction algorithm, also it is mostly used in visualization of data that has high dimension [70].
- It can interpret nonlinear relationships between features, which PCA does not perform well [77].
- It is able to preserve the local structure of the data, i.e. points that are close to each other in the high-dimensional data set will tend to be the same in the low dimension data.

### 3 Experimental Setup

In this section, the experimental setup and the ultrasonic signals are described. The experimental data used in this work were obtained previously and also used in [28, 49]. This data were provided by the authors for the development of the thesis.

The experimental setup used is shown in Fig.3.1, this is composed of a 3 mm thick, 800 mm long and 100 mm wide aluminum plate. The plate is mounted on a bridge structure, where the left end is screwed to the structure and the right end is fastened to a moving support with brackets that prevents the plate from twisting when it is subjected to different stress values. Ultrasonic waves are generated by applying an excitation signal to the left piezoelectric transducer (Tx), bonded to the plate's surface and propagate within the plate, being received at the right piezoelectric transducer (Rx) also bonded to the plate's surface. As reference, the actual strain value is measured by a resistive strain gauge (G) placed in the center of the plate.

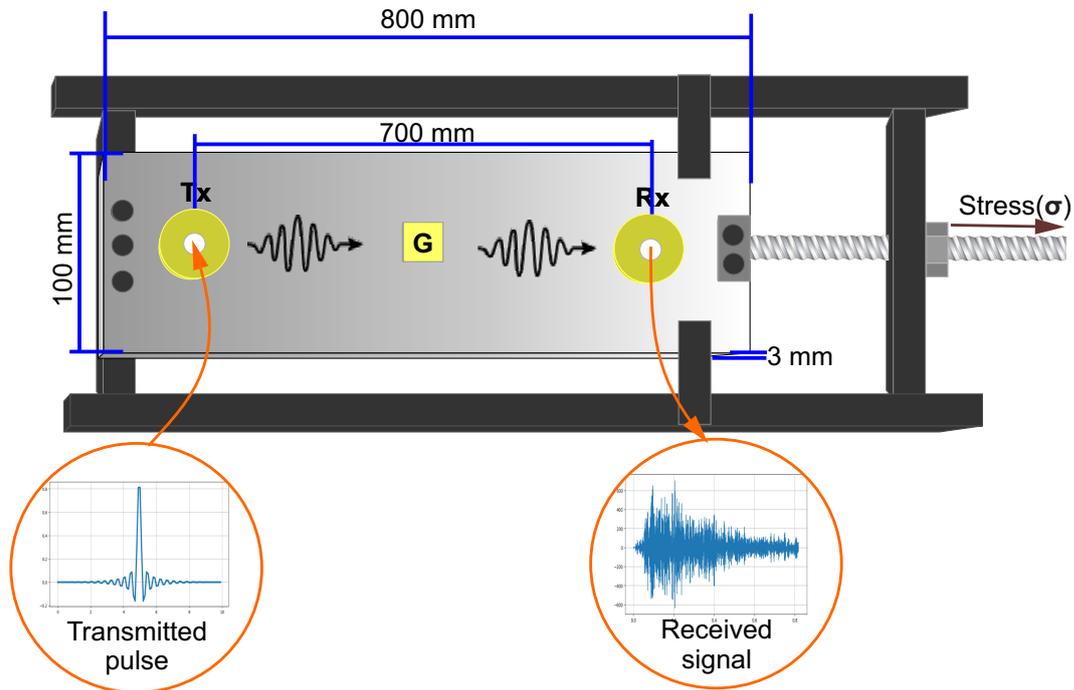


Figure 3.1: Experimental setup description. The lower plots illustrate the excitation signal (left) and the received signals (right).

The piezocomposite transducers used were constructed by cutting a piezoelectric ceramic (PZ37 from Ferropem) in one direction, making parallel bars, and filling the space between the bars with polymer, which makes this transducer

have lower impedance and a wider frequency band than the simple ceramic transducer. Table 3.1 summarizes the constructive characteristics of the transducer, showing its respective resonance frequency and dimensions [28].

Table 3.1: Transducer piezoelectric constructive characteristics

Nominal frequency (MHz)	Ceramic thickness (mm)	Element width (mm)	Polymer width (mm)	Pitch (mm)
2.2	0.65	0.1	0.04	0.14

The commercial ultrasound equipment (Open System, Lecoer Electronique, Chuelles, France) was used to generate and receive arbitrary ultrasonic signals, this equipment has a 12-bit analog to digital converter receiver. The sampling frequency was set as 10 MHz. Each run measured 8192 samples of waveform data, which amounts to 819.1  $\mu$ s of total run time. The acquisition window for the signals starts at 130  $\mu$ s, which is the time that the first wave modes arrive, and consequently ends at 949.1  $\mu$ s.

The excitation signal is a sinc like signal with a 2.5 MHz bandwidth, which is shown in Fig. 3.2. It generates ultrasonic waves in the plate and due to the broadband excitation signal, several guided waves modes propagate within the plate and are then received at the opposite transducer.

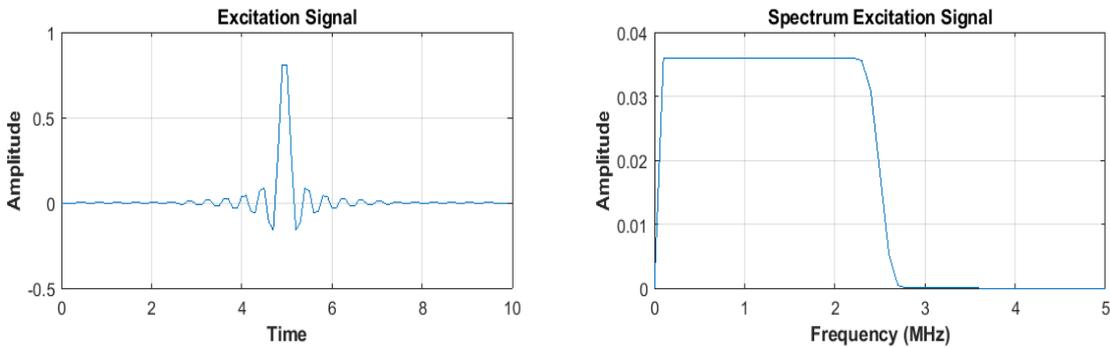


Figure 3.2: Excitation signal (sinc like signal with 2.5 MHz bandwidth) and Spectrum.

### 3.1 Measurement Campaign

The measurement of received signal was carried out as follows. The excitation signal were applied to the transducer Tx, which generated guided

waves in the plate. Those propagate and were received by transducer Rx, as depicted in Fig. 3.1. Stress was applied in discrete steps, while emission and reception of the ultrasonic waves were carried out. This loading and acquisition process was repeated four times. In each of them, the stress levels could be different, as well as the number of acquired signals samples in each stress level. In total, 499 different runs were collected by varying the applied stress.

The received signals are shown in Fig. 3.3, which are identified by colors related to the level of stress applied on the plate, with colors varying from red to dark blue. These signals are composed of different modes of guided waves created within the plate structure, as seen in the dispersive curve Fig. 2.2, and reflections, which mix in the time-domain. These reflections are caused because the transmitter and receiver transducers are positioned close to the plate's end. The superposition of the signals at the reception renders received signal interpretation complicated, as can be seen in Fig. 3.3.

It is worth highlighting that no pre-analysis of the sensitivity of predominantly generated modes, due to the excitation signal, were performed since it was adopted here a purely data-driven approach.

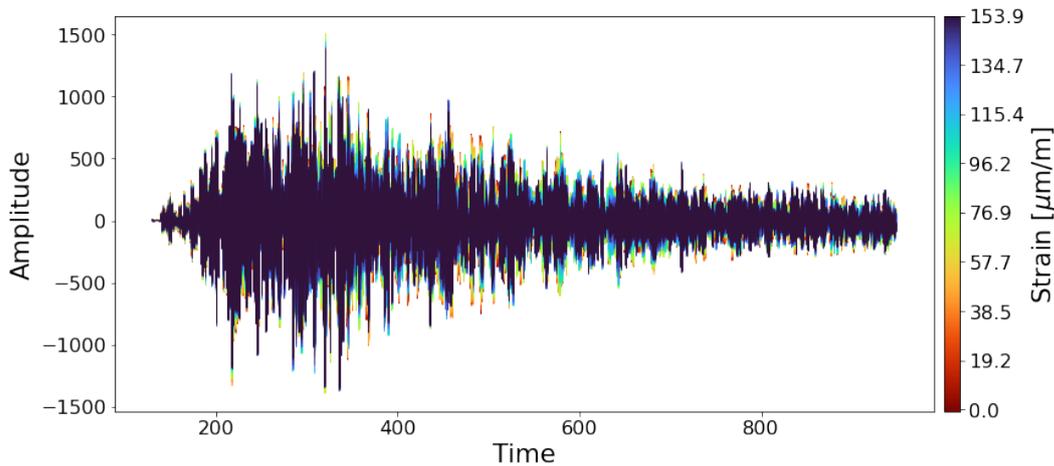


Figure 3.3: Visual representation of all received signals with colors associated with the strain perceived on the plate. All received signals refer to the same time instant relative to the transmission.

Fig. 3.4 shows the received signal in more detail by dividing the time axis into shorter windows. As can be seen, the difference is quite subtle and is more evident in some time instants. Additionally, due to the aforementioned stress-dependence behavior, where different modes exhibit distinct sensitivity, one can notice that the waveform variation is more evident in some time instants.

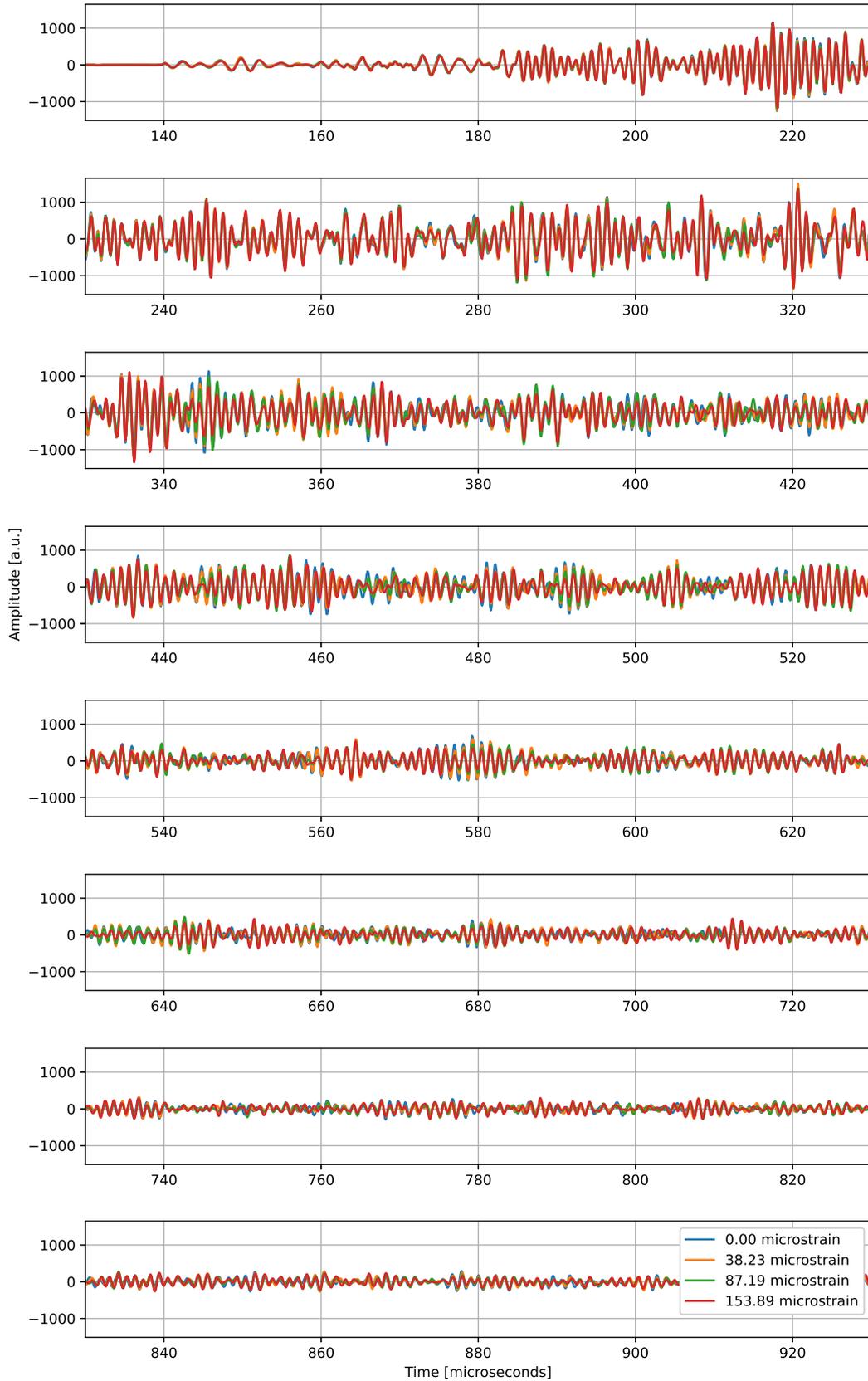


Figure 3.4: Plot of four waveforms received under different stress conditions.

In Fig. 3.5 shows the histogram of the stress values measured throughout the

test campaign, which spans from 0 to about  $150 \mu\text{m}/\text{m}$ , which corresponds to about 10MPa of tensile stress.

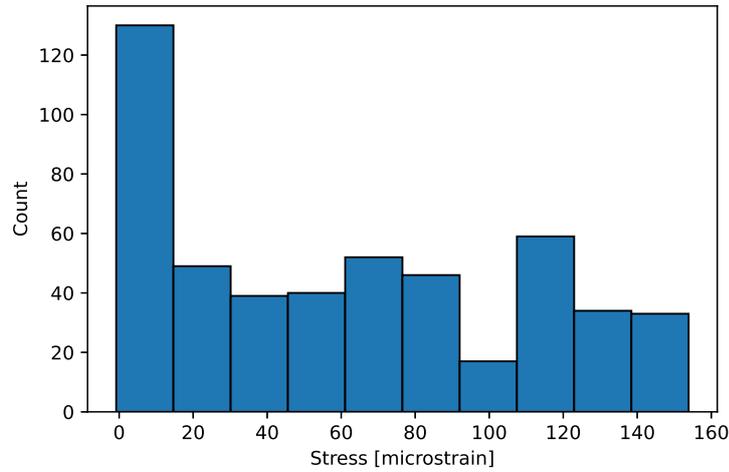


Figure 3.5: Histogram of the stress applied during all static experiments.

As can be seen, the difference between received signals is quite subtle due to the aforementioned stress-dependence behavior, being more evident in some time instants. These differences make the use of ML models possible, as will be shown in the following chapters, since these models are able to identify such differences which could not be easily perceived by human analysis.

## 4

# Improved Stress Estimation with Machine Learning and Ultrasonic Guided Waves

This chapter presents the results of the supervised approach, they correspond to the contribution of the published paper entitled: 'Improved Stress Estimation with Machine Learning and Ultrasonic Guided Waves' [18]. This section is divided as follows: Section 4.1 presents an exploratory analysis of the ultrasonic signals for each stress value; Section 4.2 presents the feature extraction procedure using PCA; Section 4.3 presents the creation of image patches for each measurement as inputs of the CNN; Section 4.4 presents construction and validation of the shallow models; Section 4.5 presents the results obtained by shallow models and PCA; Section 4.6 presents a brief discussion of the topic.

In the present chapter, the focus is on mapping the measured signals of ultrasonic guided waves traveling through the medium to the stress to which it is subjected. Thus, as explained in Section 2.3,  $X$  is constructed according to the waveforms, using dimensionality reduction for the shallow models, and  $y$  is the respective stress measurement. The interest here lies in building a predictive model that may be then deployed to perform stress estimation with guided waves measurements.

### 4.1 Exploratory Analysis

The measurements of various stress conditions are analyzed in the following. The resemblance and consequent difficult interpretation of the raw waveforms detailed in Fig. 3.4 show that they carry information regarding the stress, at certain time intervals, which differ depending on the stress amplitude, in a nontrivial way. This is due to the different stress-dependence of the several guided wave modes within the received signal, according to the acoustoelastic effect, which is inherently low sensitive in metal [22]. Additionally, due to the complicated nature of the received signal in the present case, with potential several modes and reflected waves. One can also see that the measurements are most of the time similar, without regards to the stress applied, as the differences occur at certain time chunks. Hence, the task at hand is non-trivial, in which low-stress variation is distributed within the whole waveform. This information is confirmed in Fig. 4.1, which shows the cross-correlation matrix when all the measurements are pair-wise compared.

The Pearson correlation coefficient is defined as [80]:

$$r = \frac{s_{XY}}{(s_X)(s_Y)} \quad (4-1)$$

where  $s_{XY}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$  the sample covariance between  $X$  and  $Y$ , and  $s_X$  and  $s_Y$  the sample standard deviation  $X$  and  $Y$ , respectively. The correlation indexes are close to unity most of the time, denoting the similarity of the signals measured under different stress conditions. Note in this figure that the stress measurements are sorted in ascending order, so that it is clear that the Pearson value (defined in Eq. 4-1) decreases monotonically when the stress is increased. It is shown, thus, that the stress is directly related to the cross-correlation of the raw waveform signals. Therefore, despite the small magnitude of stress and the intrinsic low-sensitivity nature, the signal indeed carries information about the applied stress.

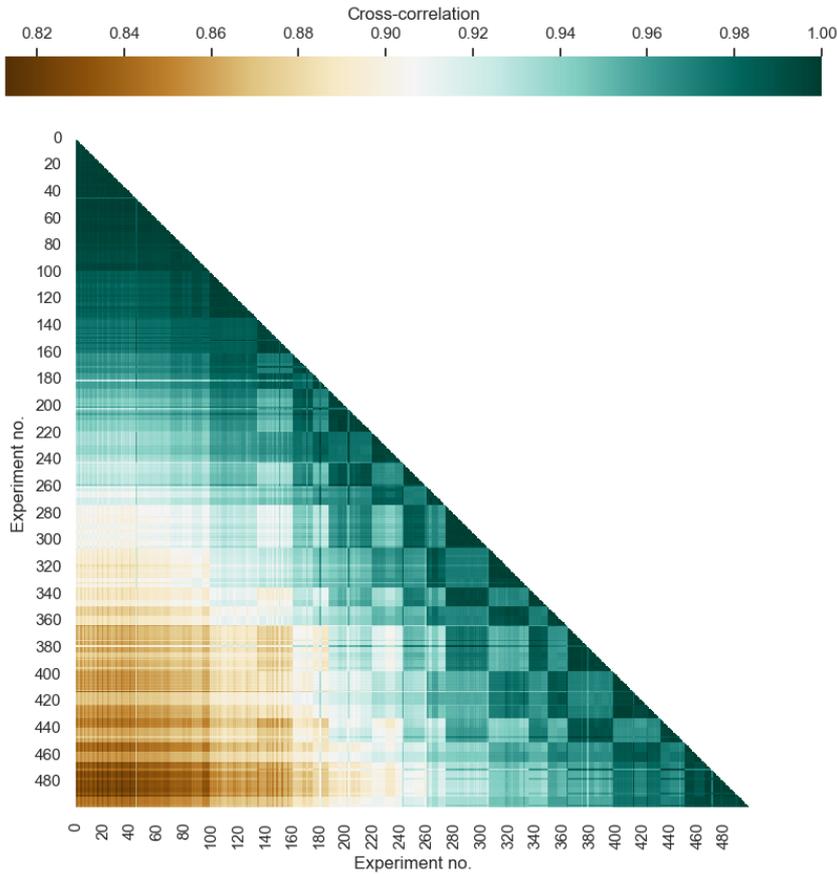


Figure 4.1: Cross-correlation matrix of all measurements made.

It was seen thus that the analysis of the waveforms is nontrivial to bare eye and that the waveforms measurements are correlated with the stress applied. The challenge for an online monitoring tool, though, remains in the creation of data pipelines that will ensure fast and accurate predictions. The first step towards the creation of a model is the feature extraction step, which is devised

next. It is worth mentioning that this is a crucial part in order to reduce the dimensionality of the model and retain only the components that are most sensitive to stress.

In the present thesis, it is important to automatically select the parts of the waveforms that are not redundant for stress estimation and will help to drastically reduce the size of the shallow predictive models. It is for this reason that it was decided to use PCA, that is important to hinder the difficulties posed by the curse of dimensionality problem, for details see e.g. [53]. Moreover, as the waveforms have collinear parts in different stress conditions, dimensionality reduction will avoid unstable parameter estimates in the model construction optimization procedure for shallow models [61].

## 4.2 Feature Extraction using PCA

The feature extraction process is related to the creation of useful and non-redundant information that can be fed for the construction of predictive models. Principal component analysis is used to this end, directly on the matrix created with all waveforms. Note that this has been obtained without using any information of stress, thus this process is referred to as an entirely data-driven procedure.

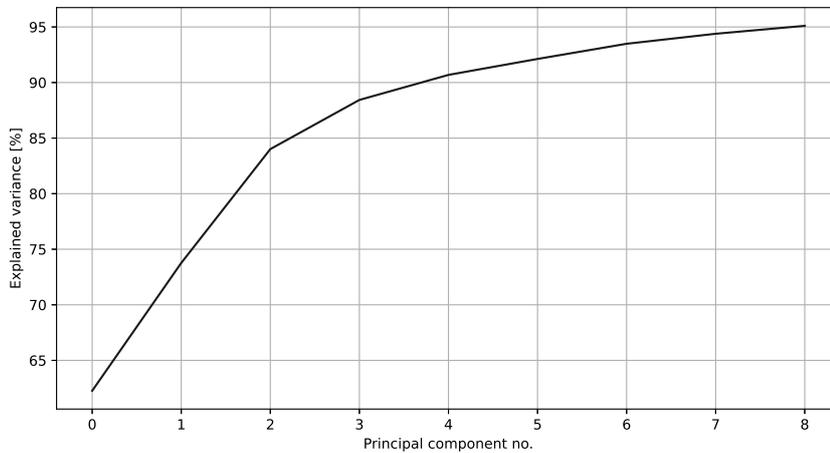


Figure 4.2: Cumulative explained variance of the principal components. Only 9 out of almost 500 components are sufficient to obtain 95%.

Having 499 measurements with 8192 samples each, a 499 x 8192 matrix is created, where each line represents a single experiment run with a different stress configuration. The cumulative explained variance up to 95% of the principal components is shown in Fig. 4.2, which turned out to be 9 for the present dataset. As seen previously, it is known that only a few time intervals are important to detect stress changes. The principal component analysis

allows automating the interval selection process that would be required if one would like to analyze the waveforms directly, in a data-driven and automated fashion. The use of such a method is important in the context of predictive modeling for monitoring applications, as it allows optimizing the inference process and build more effective models, as will be seen in the results.

Figure 4.3 depicts the normalized principal components obtained when it consider up to 95% of cumulative explained variance, normalized in the range  $[-1, 1]$ . Observe that, in contrast with Fig. 3.4, all dimensions are sensitive to the stress applied to the plate. Such dimensionality reduction is important since unimportant features will deteriorate the learning process, [81] and have made it possible to greatly improve both the accuracy and memory use of the models.

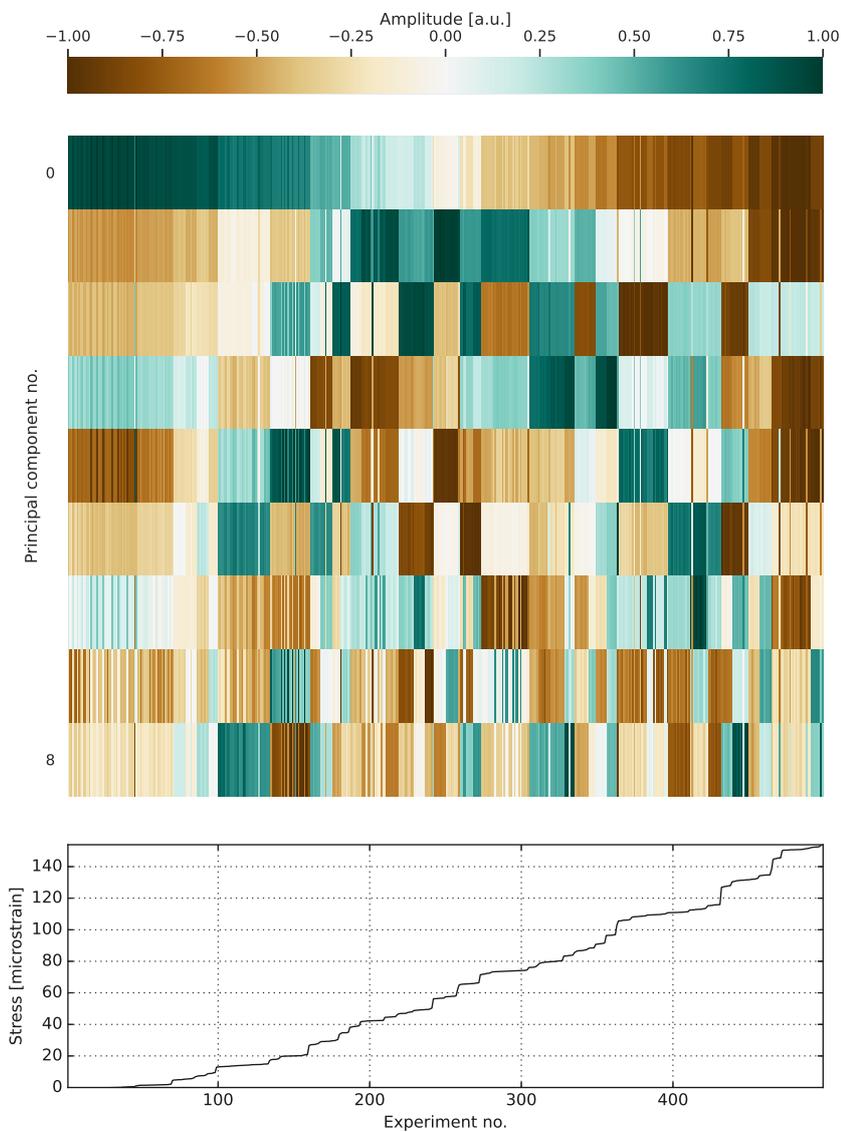


Figure 4.3: Principal components obtained from all measurements made (upper), ordered by ascending order of stress applied (lower).

### 4.3 Inputs for the Convolutional Neural Network

One of the great advantages of CNNs is that they do not require any feature extraction process. For validation, the results are compared with [34] and for that end, the same modeling procedure they did is implemented here so that fair comparison is possible.

Therefore, image patches were created for each measurement. The procedure was carried out as follows: each measurement originally acquired vector is reshaped into a squared single-channel image. A pre-processing step was necessary for our case, as the number of samples was 8,192, which is not square-rooted to an integer, so the measurements are linearly interpolated so that the final measurements are duplicated to a total of 16,384 samples. This extended and interpolated waveform was then reshaped to a 128x128 image, which can then be fed to a CNN. Fig. 4.4 shows some examples of the image patches constructed on the basis of the guided waves measurements.

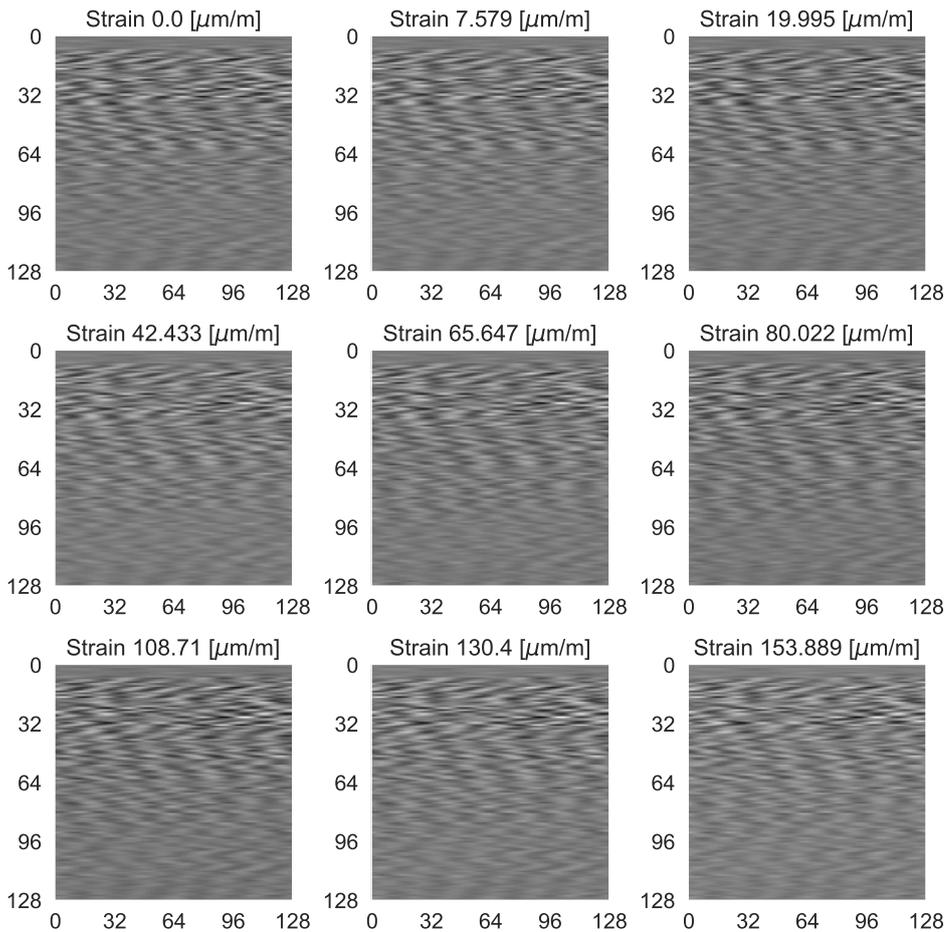


Figure 4.4: Waveforms reshaped to images that can be fed to a CNN, by varying the magnitude of stress applied.

It is possible to see that the waveform images are quite similar, turning the es-

timation of stress based on this quantity nontrivial, as shown in Fig. 3.4. This fact justifies the construction of the models that can ideally perform stress estimation from nontrivial patterns automatically by ingesting and relating input-output tuples in a mathematical mapping.

In comparison to the construction procedure for shallow models, it is highlighted that the CNN uses a higher dimensional input vector, doubled in terms of the number of samples. This procedure was followed for the following reasons.

- The CNN architecture was built to deal with squared-sized images, as in [34], and the same was done for fair comparison;
- Decreasing the number of samples to 4,096 (64 x 64 image patches) would involve smaller Nyquist frequencies;
- Applying PCA to the interpolated waveforms ends up at the same reduced-dimensions matrix if compared to the original waveforms. The explained variance was virtually the same and when each element of the reduced-order matrices were compared the  $R^2$  coefficient was very close to unity. The reason is that the interpolation adds even more redundant information for the feature reduction process, which ends up at the same reduced input matrices when using PCA. Hence, there is no difference in building a model with the reduced-order inputs using 8,192 or 16,384 samples. The simpler procedure with the original waveforms was opted for the construction of the shallow models.

#### 4.4

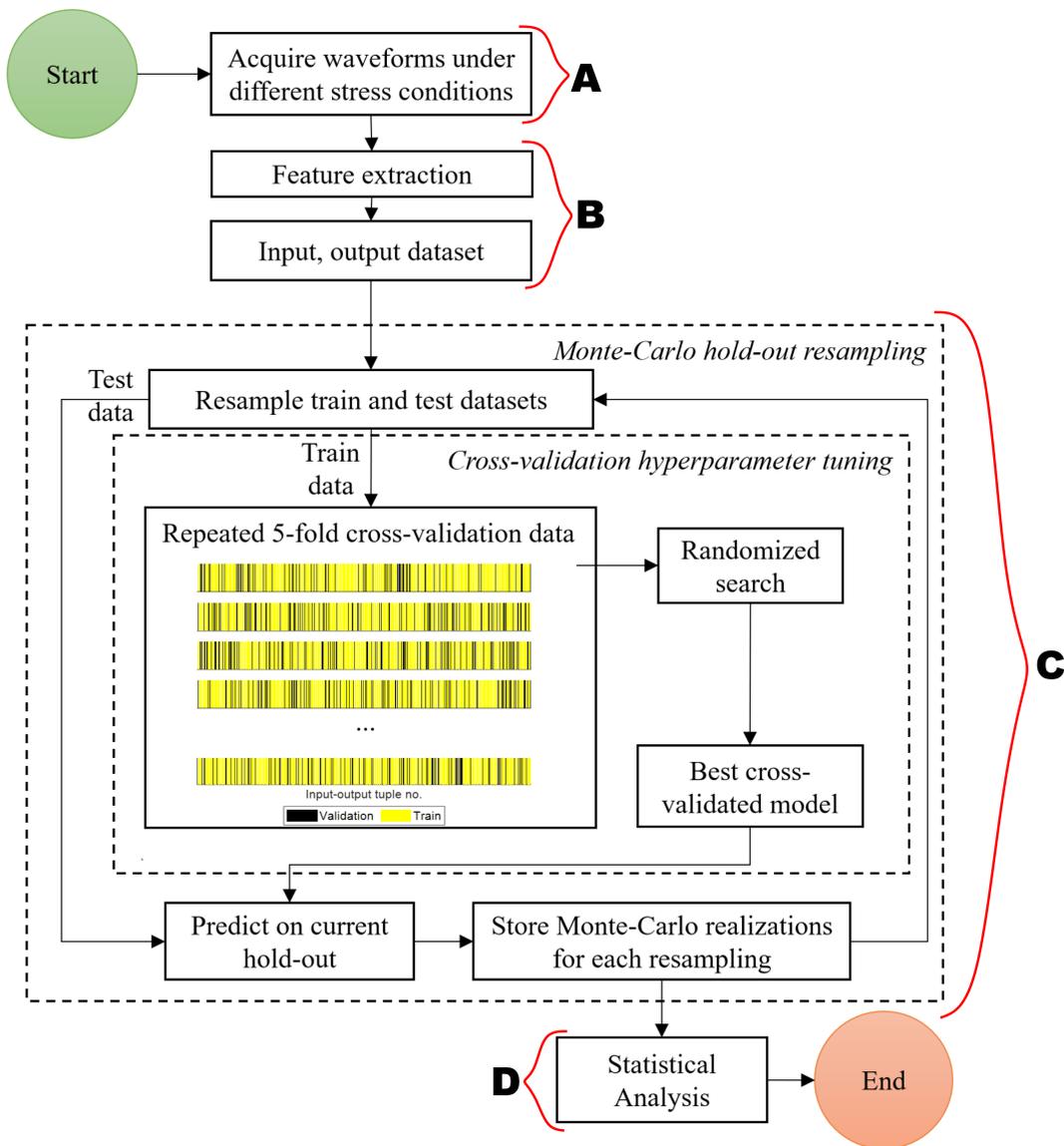
#### Resampling-Based Model Construction and Validation

The model construction and validation procedure is made with a repeated 5-fold cross-validation scheme for defining the hyperparameters and a Monte-Carlo holdout model test [82], as summarized in Fig. 4.5. The procedure is detailed as follows:

- (i) **A:** In this part the acquisition of waveforms under different stress conditions is performed, this procedure is described in the Chapter 3.
- (ii) **B:** The signal is conditioned for feature extraction based on PCA which then generates the whole input-output dataset pairs.
- (iii) **C:** The overall dataset is resampled many times to be randomly split into train and test datasets with different realizations. The training dataset is used for constructing the models, which are evaluated against many realizations throughout the repeated 5-fold cross-validation procedure.

Typical hyperparameter distributions are set for the model and, for each set of sampled hyperparameters, the best one is taken, according to the cross-validation, to analyze its performance in the test dataset. This procedure is repeated many times in a Monte Carlo evaluation. During the repeated 5-fold procedure, the hyperparameters are optimized using a randomized search procedure [83]. Such a procedure allows evaluating how the model performs in many different train/test combinations, having the hyperparameters tuned with cross-validation.

- (iv) **D**: MSE values, model sizes and hardware consumption are obtained for each model tested.



PUC-Rio - Certificação Digital N° 2012717/CA

Figure 4.5: Monte-Carlo hold-out approach using grid search and repeated 5-fold cross-validation.

It is important to mention that the procedure, aims at evaluating which model performs better in different regions of the holdouts, and that a final model would, ideally, be constructed using all available data. Moreover, as the measuring process is expensive, by replicating the holdout resampling enriches the operating envelope that the test set contains without sacrificing the analysis on robustness by avoiding to use few test samples.

## 4.5 Results

In the present section, it is shown the results of the predictive model construction for stress monitoring. In summary, the analysis is depicted here as follows. As detailed in Fig. 4.5, it is employed a random search procedure for evaluating several different hyperparameters on each resampling for holdout test. Then, the outcomes of each iteration are analyzed on distinct holdout sets, which enables fair comparison among different operating conditions. The model accuracy and complexity are also given to establish the best compromise between performance and resource usage. All results were generated in Python using sci-kit learn [84] for the shallow models and metrics, and TensorFlow [85] for implementing the CNN architecture used in [34] which is used in this work as the baseline.

### 4.5.1 Model Validation Settings

One hundred holdouts were used for all models using randomized resampling with 50% split. Being so, 100 realizations were analyzed of the models on different train-test dataset splits and store all holdout predictions in order to evaluate the predictive performance of the models statistically. Since there are 499 input-output pairs, the models are tested in approximately 25,000 input-output tuples. Shallow models are created using 5-fold cross-validation repeated 20 times, and the hyperparameters are defined using a randomized grid search strategy for 40 realizations of the distributions. Table 4.1 shown the all values of hyperparameters of each model, all these were used for construction of models. For details on the Monte Carlo holdout test with repeated 5-fold cross-validation, see Subsection 4.4.

The CNN is created without cross-validation, as the hyperparameters are taken from [34] for the sake of comparison. Additionally, the repeated 5-fold randomized grid search strategy would require excessive computational resources. The architecture given in [34] for stress estimation and used here for comparison is inspired in LeNet-5 architecture [86], having 2 convolutional layers with 30

3x3 filters and average pooling each, followed by 2 fully connected hidden layers with 56 and 28 neurons each. The training is run for 100 epochs using mini-batches of size 16, having learning rates set as  $10^{-2}$ ,  $10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$  successively for refining the final model. The CNN models use the 128 x 128 image-based features as described in Section 4.3. All the other models tested, that is, the shallow models, use the features obtained by the PCA feature dimensionality reduction method given in Section 4.2. The Fig. 4.6 shows that only the shallow models were constructed and validated using a Monte-Carlo hold-out procedure, unlike for the CNN where it was created with architecture and parameters given of [34].

Table 4.1: Hyperparameter settings for the models tested in the present work using repeated k-fold cross-validation and randomized search.

Model	Hyperparameter	Distribution/option
LRG	$\alpha$	Log-uniform [ $10^{-1}$ , $10^3$ ]
SVR	C	Log-uniform [ $10^{-1}$ , $10^3$ ]
	$\gamma$	Log-uniform [ $10^{-4}$ , $10^0$ ]
	$\epsilon$	Log-uniform [ $10^{-4}$ , $10^1$ ]
	Kernel	Linear, polynomial, radial basis functions, sigmoid
	Degree (polynomial kernel)	Uniform integer {2, 3, ..., 5}
DTR	Criterion	MSE, Friedman MSE
	Splitter	Best, random
	Maximum features	Uniform [0, 1]
kNN	No. of neighbors	Uniform integer {2, 3, ..., 100}
	Weights	Uniform, distance
RFR	No. of estimators	Uniform integer {2, 3, ..., 100}
	Criterion	MSE, MAE
	Maximum features	Uniform [0, 1]

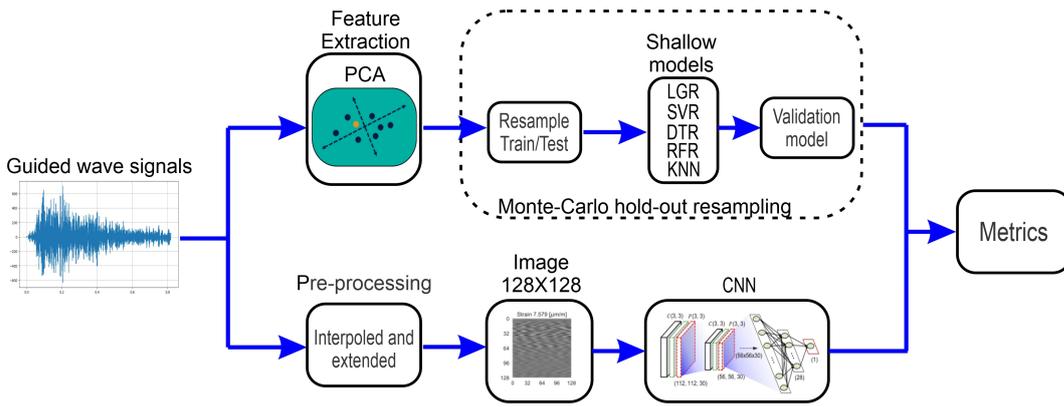


Figure 4.6: Summary of the workflow for the construction and validation of the supervised models.

### 4.5.2 Error Analysis

The precision of the models obtained for stress estimation is investigated hereafter. The quantitative error prediction results in holdout are presented in Table 4.2, which is ordered in ascending magnitude of MSE.

It is possible to see that all models have obtained reasonable median values, and that the best model in terms of overall MSE is the SVR. CNN, which is the second-best model by MSE, obtains an increase of 9% compared to the MSE value of SVR, as shown in the last column of Table 4.2. The SVR associated with PCA feature extraction can improve the MSE metric by a considerable margin, reinforcing the importance of efficiency both in terms of accuracy and model construction time and inference, as the SVR is considerably faster to build and run than the CNN.

Table 4.2: Squared error statistics of holdout realizations for all models tested, ordered in ascending order of MSE.

Model	Mean	Standard Deviation	Min.	1st quartile	Median	3rd quartile	Max.	% MSE increase
SVR	10.33	57.77	0.00	0.11	0.55	2.32	922.72	-
CNN [34]	11.32	62.14	0.00	0.22	1.05	3.46	908.05	9.56
RFR	11.72	61.79	0.00	0.04	0.22	1.12	1103.29	13.47
KNN	11.94	67.05	0.00	0.04	0.21	1.00	1079.43	15.56
LRG	12.62	61.96	0.00	0.34	1.62	5.74	902.45	22.13
DTR	18.01	144.68	0.00	0.02	0.12	0.43	8988.69	74.32

The information of the errors in holdout is given in Fig. 4.7, where raincloud plots [87] of the errors are shown, which are represented in ascending order

from the largest to the smallest MSE value. Those show the distribution of the errors in all holdouts. Apart from the allusions made before concerning the MSE, it is also added that the models have similar distributions and there is a considerable amount of large errors made by the worst models in terms of MSE, as it is shown in Fig. 4.7.

Also, interestingly, the models perform differently in sections of error amplitudes. This may shed light on how to further improve the stress estimation procedure, as in future work one might implement a final model that interprets the outputs of all models so that an even more accurate stress estimation method is devised.

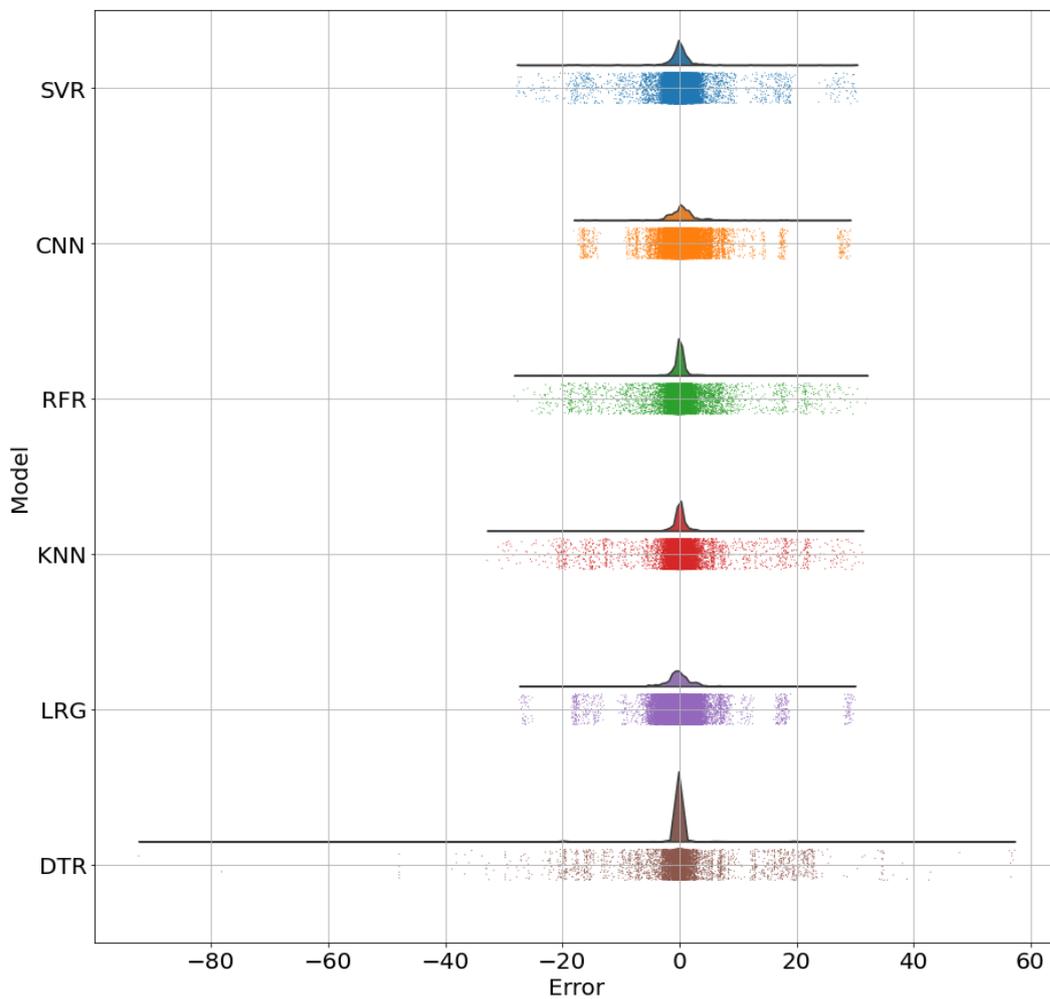


Figure 4.7: Distribution of the error for all holdout model predictions, in ascending order of MSE (upper to lower). It is interesting to note that the models perform differently depending on the stress condition.

The stress predictions versus the real measured values are given in Fig. 4.8. One can readily relate it with the information given both in Table 4.2 and Fig. 4.7. It is possible to see that the SVR model can provide accurate results in terms of stress estimation using the PCA feature space reduction technique

and improved error metrics when compared to previous work published [34]. In the following, the analysis is extended not only to precision but also to model size that is generally related to inference time, model creation/update computational resources, and hardware consumption for real-time use.

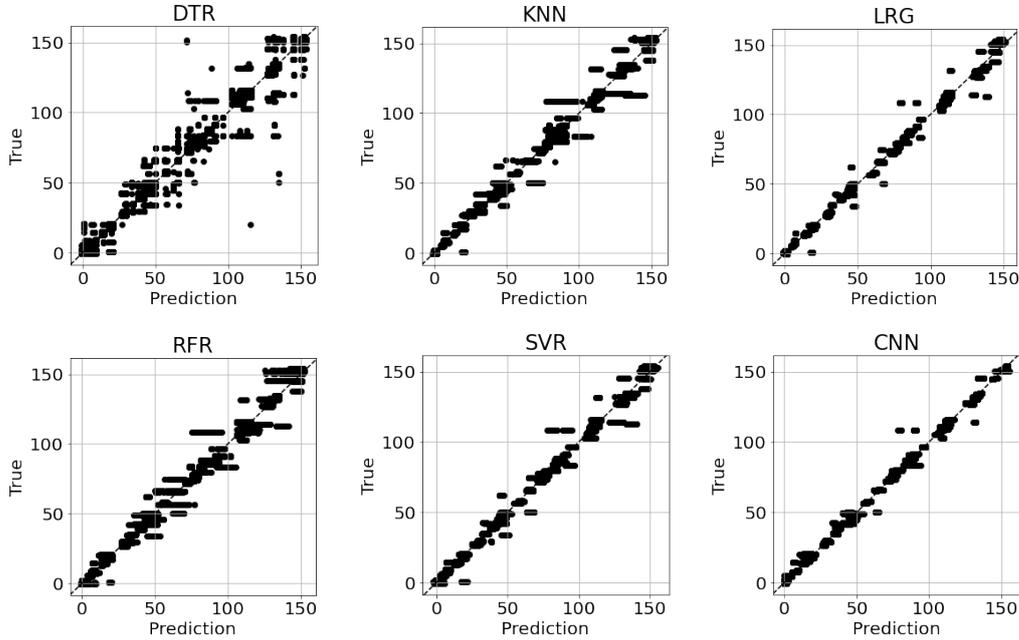


Figure 4.8: Predictions made on all realizations for different holdout cases, organized by different model classes.

### 4.5.3 Model Size Analysis

In order to evaluate the model size, the hard disk space is measured in bytes taken to store all variables needed to perform inference. Table 4.3 summarizes the numerical values in bytes of all models tested. The LRG and CNN are highlighted, since they do not change their sizes throughout each holdout because LRG size is defined by the number of inputs, which is fixed to the resultant number of principal components, and the CNN size is the same as in [34]. As it is an instance-based model, KNN has the same size for almost all holdouts, the only difference here is due to the random non-exact 50% split of 499 input-output tuples. In this table, it can be seen that SVR, which is the most accurate model proposed herein, takes 2 orders of magnitude less space to store than CNN, which is the second most accurate model. It represents a great advantage of the modeling workflow herein presented. The smaller model is the LRG, however, the cheapest model to embed implies 22% greater error as shown in Table 4.2. The SVR and LGR present the best trade-off

among precision and hardware consumption, and both use the same modeling philosophy based on PCA proposed originally in this paper.

Table 4.3: Statistics for the model size in bytes for the models tested in the present work using repeated k-fold cross-validation and randomized search. Please note that LRG and CNN do not vary their sizes in each holdout resample.

Model	Mean or actual value	Standard deviation	Min.	1st quartile	Median	3rd quartile	Max.
LRG	1,083	-	-	-	-	-	-
SVR	18,647	5,360	5,716	16,470	21,802	22,371	22,435
KNN	26,179	0.20	26,179	26,180	26,180	26,180	26,180
DTR	33,139	132	32,824	33,080	33,208	33,217	33,347
RFR	1,503,233	375,483	473,854	1,266,619	1,576,439	1,796,660	2,014,926
CNN [34]	6,110,720	-	-	-	-	-	-

Such trade-off can be seen if one compares the MSE and the mean size in bytes of the models, as depicted in Fig. 4.9, which is hereafter referred, the model construction precision-size decision space. Here it can be seen that the precision-size decision space can be divided into Pareto-optimal solutions and dominated solutions [88] and, in this case, the SVR and LRG models dominate all other models being a better design decision in terms of error and hardware consumption. Black dotted lines denote the region, in this space, that the two best models perform better with respect jointly to model performance and size, i.e. SVR presents the minimal error and LRG presents the minimal hardware consumption of the all models.

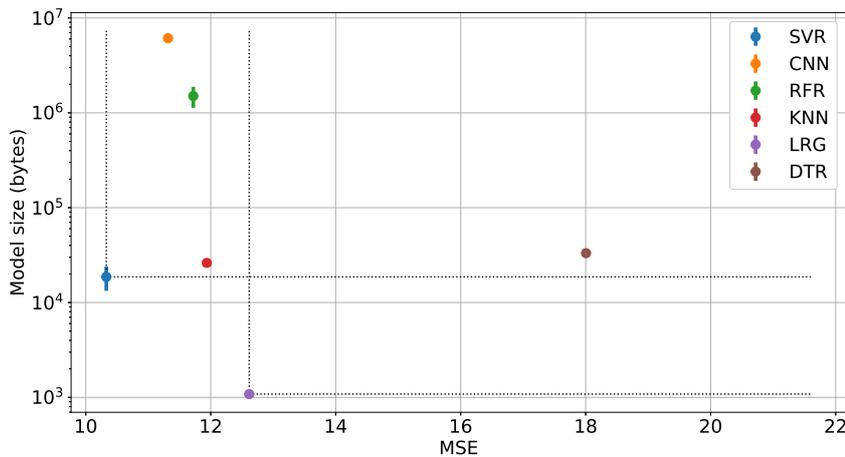


Figure 4.9: Model sizes, represented by mean and standard deviation, and respective MSE metrics in various holdout phases. Black dotted lines denote the region, in this space, that the two best models perform better with respect jointly to model performance and size.

## 4.6

### Discussion

The supervised approach was inspired by the results obtained in [34] using CNNs that are aimed towards the joint investigation of accuracy and overall model complexity. When compared to previous work, it devised a different feature extraction method that enabled us to, not only produce more accurate predictive models but also to dramatically decrease the model size. The best models found in terms of accuracy and model size are SVR and LRG, in strong contrast with previously reported results in [34] based on CNNs for the process of stress estimation, as it has been demonstrated.

It is recognized that the CNNs have powerful data-driven automatic feature extraction mechanisms and, maybe going through an exhaustive architecture search [89], one might find a better network than the one used in [34] which had their results reproduced here using the signals of this thesis. However, non-destructive evaluation methods that use ultrasonic guided waves as an indirect measuring principle, such as data-driven stress estimation, are characterized by real-time use, sampling at high-frequency rates, and distributed sensing. In such a scenario, naturally, the accuracy is important, but so is the model size and its related complexity, as a large enough model may produce unfeasible computational implementations. From the workgroup experience in embedding ANNs on hardware [90], even small networks can occupy considerably large chunks of expensive hardware and consume scarce power in such scenarios. In this regard, it was also tested feedforward architectures with up to 8 layers and 16 neurons each with PCA feature space reduction, obtained MSE of 18.55 which is greater than the last model show in Tab. 4.2. For this reason, feedforward architectures were not considered in this thesis, because they are no superior to the CNN architecture from [34] in terms of model accuracy.

In the literature, the embedding of machine learning models in FPGA has seen an increased interest in the community (see e.g. [91–95] to cite a few), fueled mainly by the internet of things in the past decade or two [96]. Such embedded implementation is still necessary for a successful practical application to analyze model complexity. As this is a fast-paced moving research field, soon it might be possible to efficiently embed such huge deep neural network model, which will enable optimized CNNs for stress estimation using fewer hardware resources. In any case, as of now, the current practice will not be able to move from conceptual models to real-world applications if this aspect is not considered throughout the model design and data ingestion process, as proposed in the present thesis.

Lastly, it was shown that the models perform differently in regions of error

amplitude. This fact can be explored in the creation of novel data ingestion workflows, having in mind that the models produce uncorrelated errors, using stacking ensembles for example. When compared to other signal processing approaches, such as the use of a reference null-stress signal to perform a time-reversed excitation [28], the use of the present data-driven approaches allows more assertive information for monitoring and the overall decision-making process, as it directly maps waveform to stress estimates.

In this thesis, ML algorithms were applied to estimate the mechanical deformation applied to an aluminum plate. Thus, a working path is proposed for the use of supervised learning, using a stage of extraction of characteristics of the data, and then entering them as input for the supervised models. A comparison was carried out using metrics such as the MSE and the size of the model, which showed an improvement in performance with the SVR model when compared to the use of the CNN architecture that is available in the literature [34].

## 5

# Unsupervised Tensile Stress Estimation Using Ultrasonic Guided Waves Signals in Plates

The present chapter presents the result of the unsupervised approach, that corresponds to PCA and t-SNE methods reported in the contribution to be submitted entitled: 'Deep Unsupervised Tensile Stress Estimation Using Ultrasonic Guided Waves Signals in Plates'. This section is organized as follows: Section 5.1 presents the feature extraction procedure using t-SNE; Section 5.2 presents the results obtained using the PCA and t-SNE methods; Section 5.3 presents a brief discussion of the topic.

### 5.1

#### Feature Extraction using t-SNE

For the t-SNE approach, the dimensionality reduction was made through *TSNE* function also from *Sklearn* package for Python [84], considering the parameters: 4 embedded space components, perplexity of 50, 1,000 optimization iterations, learning rate of 10 and exact method for gradient calculation. Therefore, the t-SNE feature extraction produced a dataset of  $499 \times 4$  reduced dimensions.

### 5.2

#### Results

In this section, the results achieved by each unsupervised learning technique used in this thesis are shown, regarding the ability to extract representative features to feed a k-means clustering. For all approaches, the unsupervised tensile stress estimation through k-means clustering was made with *KMeans* function from *Sklearn* package for Python [84], considering as parameters: 3 clusters, 100 centroids configuration explored up to 400 iterations, being these initialized by *k-means++* method. Fig. 5.1 shows an overview of the unsupervised stress estimation frameworks used for data labeling. Three phases are shown and explained as follows:

- **Data acquisition:** Acquisition of the ultrasonic waveform, as described in chapter 3
- **Features Extraction:** PCA and t-SNE methods were used to find characteristics associated with the stress state of the plate. In order to perform the PCA, raw signals were considered. It was decided not to apply

any pre-processing technique, seeking to avoid any data mischaracterization. However, having obtained the principal components, the reduced dataset was normalized between  $[-1, 1]$  and for performing the t-SNE, a preprocessing technique was employed, where the raw data set was normalized between  $[-1,1]$ .

- **Clustering result:** The k-means clustering algorithm was used for data labeling and was performed using the extracted features to feed k-means, where three clusters could be perceived.

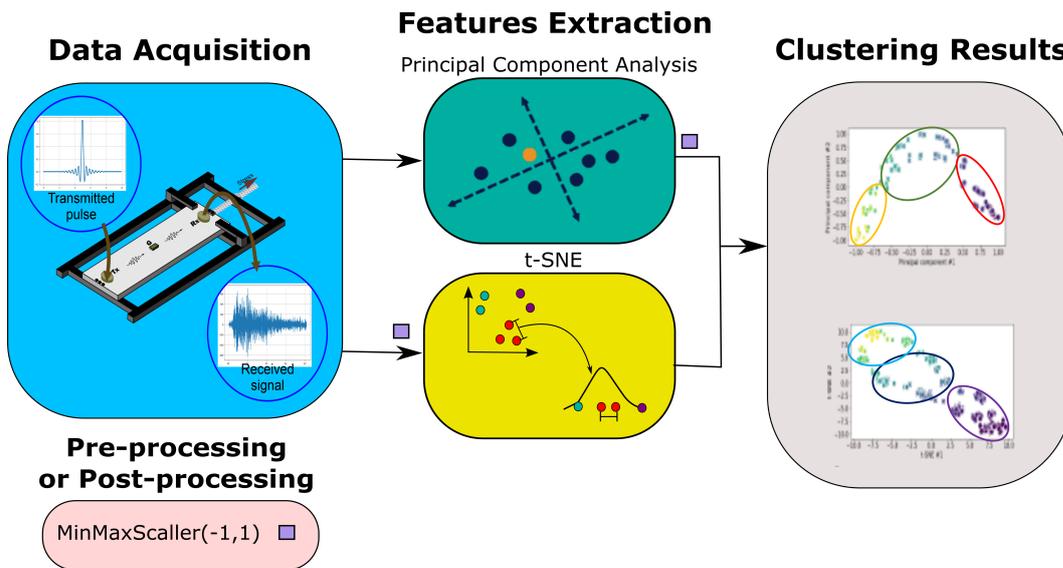


Figure 5.1: Overview of the unsupervised stress estimation frameworks used for data labeling

Table 5.1 shows the result of cluster evaluation metrics. These were used to evaluate clustering performance and when the ground truth labels are not known. Cluster evaluation metrics were explained in Section 2.4 and summarized below for convenience.

- **Silhouette Coefficient (SC):** is a metric that analyzes how well a point fits into a cluster. The value of SC ranges from  $-1$  to  $1$ . If coefficient close to  $+1$  means that the clusters are well separated from each other, but if coefficient close to  $0$  means that it is close to the cluster boundary, and finally if coefficient close to  $-1$  means that clusters are assigned in the wrong way.
- **Calinski-Harabasz Index (CH):** evaluates the cluster validity based on the sum of the inter-cluster and the intra-cluster dispersion for all clusters. A higher value of the CH index means that the clusters are dense and well separated, although there is no “acceptable” cut-off value.

- Davies-Bouldin Index (DB): The goal of DB is to maximize the distance between clusters and minimize the distance between points in a cluster. The closer to zero its value, the better the partition.

Table 5.1: Clustering evaluation metrics.

Approach	SC	CH	DB
PCA	0.258	103.721	1.907
t-SNE	0.462	923.679	0.866

### 5.2.1 PCA approach

For the PCA approach, the feature extraction was performed in order to obtain 95% of explained variance, which totaled 9 principal components as described in Section 4.2. Table 5.1 shows that the PCA approach presented the worst result, for having SC farthest from 1, the smallest CH and DB farthest from zero. The first two principal components obtained by applying PCA were used to project the data onto a two-dimensional plane and to observe the clustering performed by the K-means algorithm, as shown in the following figure:

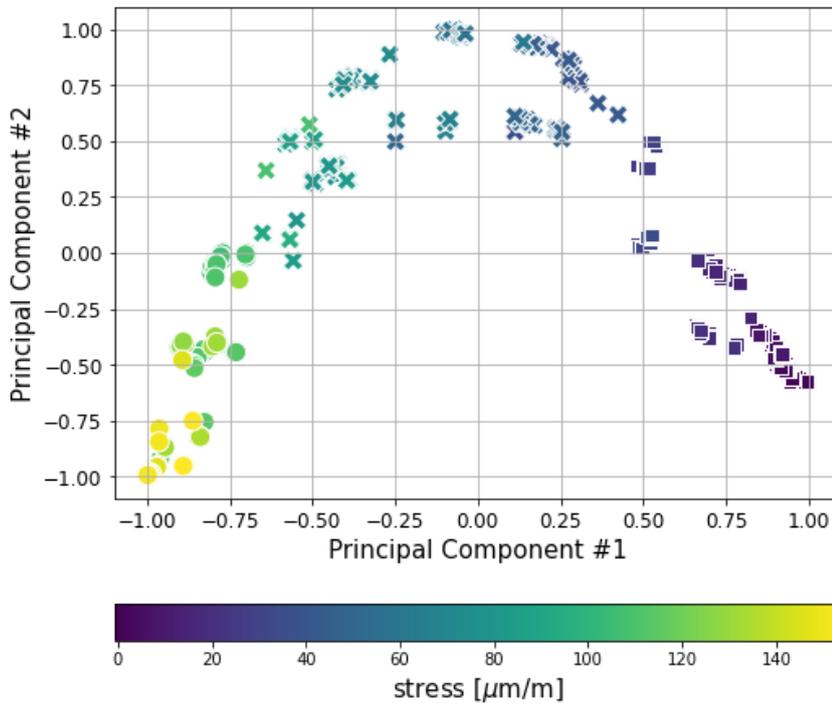


Figure 5.2: K-means clusters represented in the space composed by first and second features derived from PCA, where ○ is high stress, × is medium stress and □ is low stress.

A visual analysis driven by Fig. 5.2 reveals that: i) the PCA clustering approach was able to sort classes with low, medium and high tensile stress groups, represented by  $\circ$ ,  $\times$  and  $\square$ , respectively; ii) the first principal component represents the microstrain variation very well, indicating an almost linear relation with the microstrain. Fig. 5.3 shows that, despite having the worst metrics, the outliers do not reach any other quartiles and the high-stress cluster presents a smaller spread.

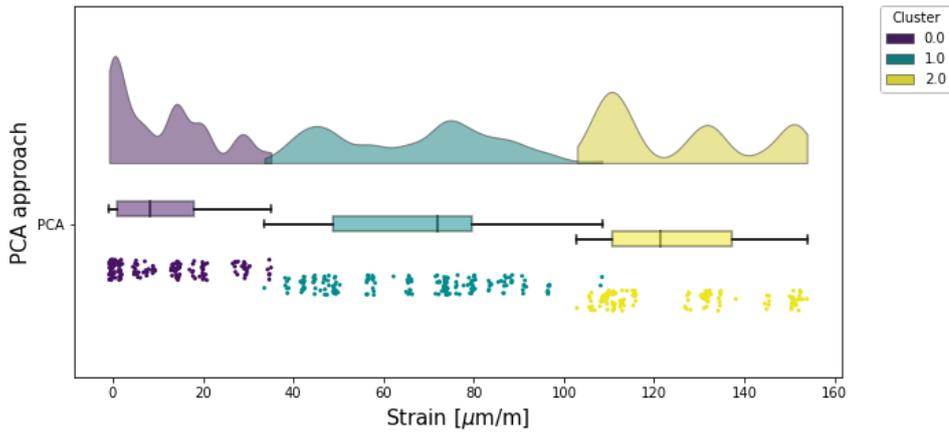


Figure 5.3: Half violin plots and boxplots for stress distribution representation of all k-means clusters derived from PCA. Purple, green and yellow colors represent cluster with low, medium and high stress states, respectively.

### 5.2.2 t-SNE approach

Table 5.1 shows that the t-SNE approach presented the SC value closer to unity, the higher CH and DB value closer to zero.

The first two principal components obtained by applying t-SNE were used to project the data onto a two-dimensional plane and observe the clustering performed by the K-means algorithm, as is shown in the Fig. 5.4, this figure reveals that: i) t-SNE approach was also able to sort classes with low, medium and high tensile stress groups, represented by  $\circ$ ,  $\times$  and  $\square$ , respectively; ii) the second t-SNE feature represents stress variation very well, indicating an almost linear relation with stress.

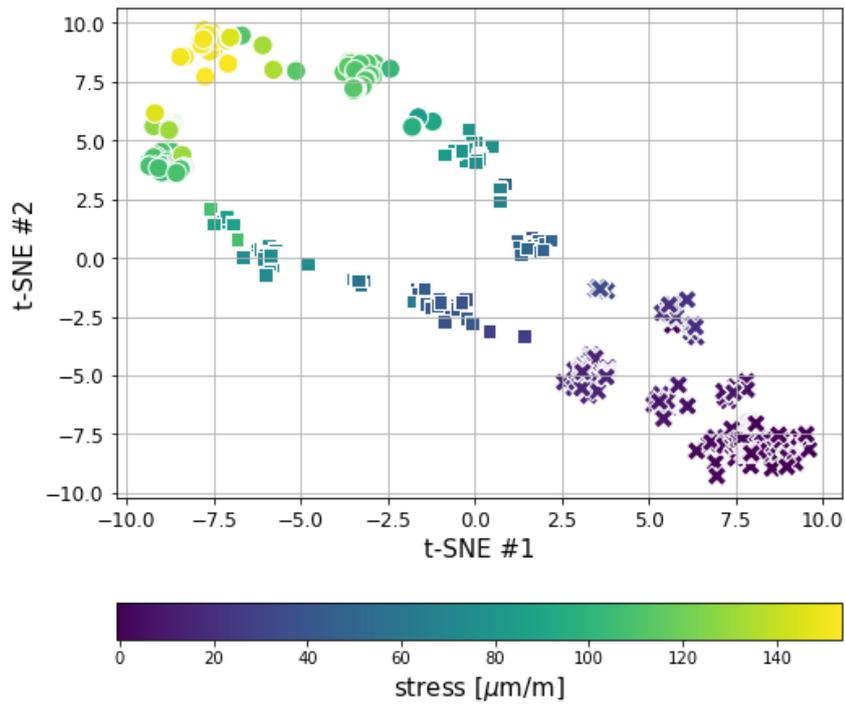


Figure 5.4: K-means clusters represented in the space composed by first and second features derived from t-SNE, where ○ is high stress, □ is medium stress and × is low stress.

Fig. 5.5 illustrates that, despite a better performance according to the evaluation metrics, the outliers almost reach other clusters quartiles and that the high-stress cluster present a higher spread, both in comparison with PCA approach.

A qualitative analysis of the results is shown in Fig. 5.2 and 5.4, where clusters are represented in the two-dimensional space composed of the first two features of each approach, where the stress is represented from high to low with a color variation from purple to yellow, respectively. Fig. 5.3 and 5.5 show the stress distribution presented by each cluster, where purple, green and yellow colors represent cluster with low, medium and high stress states, respectively.

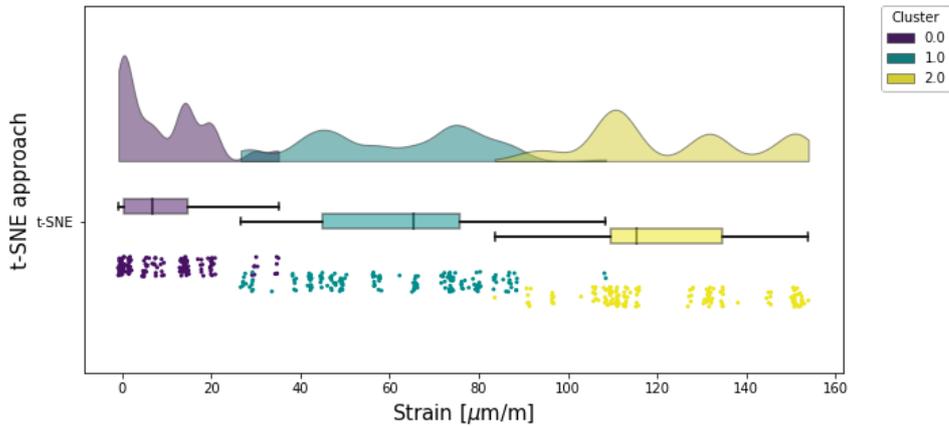


Figure 5.5: Half violin plots and boxplots for stress distribution representation of all k-means clusters derived from t-SNE. Purple, green and yellow colors represent cluster with low, medium and high stress states, respectively.

### 5.3 Discussion

The result for the unsupervised approach were presented in Section 5.2, which shows that the clustering evaluation metrics would be a good indication to define the best approach and order results from best to worst as: t-SNE, for presenting higher CH and lower DB in relation to PCA, which presented the worst value for all metrics. However, a qualitative analysis is also important, since PCA presents a greater order of magnitude for the first principal components and, when the data is scaled between  $[-1, 1]$ , the distances used to calculate the evaluation metrics become equally important.

Therefore, a qualitative analysis of the problem reveals that the PCA approach presented a less scattered high-stress cluster, which is an interest group in real applications. Furthermore, the PCA approach presented well-defined cluster boundaries, being the method in which the groups overlap less, as their outliers are closer to the clusters median. The PCA method has a simpler and less computational costly implementation, also presenting a lesser outcome variation, since Singular Value Decomposition (SVD) driven PCA is a deterministic method.

The acquired results indicate that all unsupervised features extraction and clustering frameworks proposed in this thesis may be used to fill the literature gap, being able to separate samples as low, medium and high stress groups, allowing large dataset samples to be automatically labeled and making this activity less costly for future works involving large UGW datasets.

## 6

### Conclusion and Future Works

This thesis deals with supervised and unsupervised approaches to estimate the tensile stress using guided wave signals. In the supervised approach it was shown that a model construction based on PCA and shallow learning provides better results, when compared to previous works, on the stress estimation problem using ultrasonic guided waves both in terms of accuracy and model size.

Specifically, it was proposed a conceptually different data ingestion workflow than [34], by employing dimensionality reduction together with shallow models. The models herein proposed presented an 10% improved result, in terms of MSE, when compared to [34]. This result is interesting as it also puts into context the extensive recent use of deep models for systems monitoring. The models herein proposed are also considerably more efficient in terms of memory use, having improvements up to three orders of magnitude and still delivering better predictions. Improving the model size ensures that the hardware usage for the models is also optimized for expensive and resource-constrained real-time monitoring tools, which has a major impact on the productivity and practical use of these experimental methods.

It is worth mentioning that the signals were acquired from a laboratory setup, these signals were used to train and assess the model. However, due to the nature of the received waveforms with several modes and some reflection, and the good performance of the model in stress prediction, mostly due to the dimensionality reduction, through keeping only the most significant principal components, indicates that the model should well generalize; though retrain may be necessary should the final measurement setup be considerably different if compared to the one used to train the models. Additionally, one should highlight that the present thesis was restricted to tensile stress applied along the longitudinal direction of the plate, which is a relatively simple stress state. Since the guided waves velocity variation also depends on the direction of applied stress [48] and its nature such as bending stress [97], shear or hydrostatic pressure [98], one could, in principle, use the present models to not only predict the stress magnitude but also its direction or natures. Since different stress states induce changes on the guided wave modes in quite a different way, for instance introducing strong variations at specific frequencies [98] or coupling between Lamb and SH modes, this points towards the promising possibility of using the presents models, with such a broadband excitation, to estimate the

full stress state, in addition to its magnitude.

In the unsupervised approach, a gap was found in the literature, regarding the use of unsupervised learning to label samples for stress estimation applications, within the area of SHM. Therefore, it sought to compare different unsupervised learning frameworks to extract features from UGW signals and cluster them, promoting their separation according to stress state.

The results show that by using both PCA and t-SNE in the feature extraction stage, it was possible to coherently cluster the samples according to stress state, which could be used for labeling large datasets in future work.

The comparison of the conventional dimensional reduction techniques was quantitatively and qualitatively driven, the latter being necessary to confirm the results, PCA is more efficient in grouping the samples compared to t-SNE. The results show that PCA is able to separate the groups of stresses defined as low, medium and high stress, with a small overlapping between the medium and low groups, while, with t-SNE, there is overlapping in the three groups causing the samples to be rounded. The use of PCA was highlighted for this application, which is an algebraic method with low computational cost for small data sets, while t-SNE has a much higher computational cost.

Stress estimation using machine learning has been only employed in few works and there are many important research topics that are not devised thus far in the specialized literature. The use of more efficient model construction procedures can be devised, by using different feature space reduction such as system identification [99] and unsupervised learning [100], or taking advantage of the behavior of the models analyzed herein concerning the performance on different error amplitudes for the construction of model ensembles [101, 102]. For the real-time implementation of such frameworks in dedicated embedded hardware, FPGAs [103–105] and GPUs [106–109] have delivered important results recently which may also be used in the context of stress estimation to provide feasible implementations for practical applications.

## 6.1

### Future Works

The present results suggest some step towards the implementation of a more robust and accurate real-time system.

- Use the models tested in this thesis with another data set, for example, another frequency, another distance between transducers, etc, in order to evaluate their performance and compare them with the results shown in this thesis.

- Use the spectrum of the signals, instead of the signals in the time domain, to compare the results with those obtained in the thesis, in order to find an improvement.
- The use of more efficient model construction procedures can be devised, by using different feature space reduction such as system identification [99] and unsupervised learning [100], or taking advantage of the behavior of the models analyzed herein concerning the performance on different error amplitudes for the construction of model ensembles [101, 102].
- Research based on physical phenomena, with analysis on the phase of the signals as pre-analysis to improve the results and also as justification of the results of the present work. This can be useful for generalization with signals of a slightly different nature.
- Implementation of real-time systems using dedicated embedded hardware, as FPGAs [103–105] and GPUs [106–109], have delivered important results recently. It can thus be used in the context of stress estimation to provide feasible implementations for practical applications.
- The results obtained by the unsupervised approach were produced from a small set of strain-labeled ultrasonic signals, showing that the proposed frameworks were able to coherently cluster the samples according to strain state, which may allow automatically labeling large datasets and making this activity less expensive for future work involving large UGW datasets. As discussed in [38] one of the approaches is data labeling, this work is suited to this approach which could be an important tool for SHM applications if it is further developed.

# Bibliography

- [1] A. Sofi, J. J. Regita, B. Rane, and H. H. Lau, "Structural health monitoring using wireless smart sensor network—an overview," *Mechanical Systems and Signal Processing*, vol. 163, 2022.
- [2] E. Fathalla, Y. Tanaka, and K. Maekawa, "Remaining fatigue life assessment of in-service road bridge decks based upon artificial neural networks," *Engineering Structures*, vol. 171, pp. 602–616, 2018.
- [3] X. Ma, F. Yang, J. Li, Y. Xue, and Z. Guan, "Fatigue life assessment method of in-service mechanical structure," *Advances in Mechanical Engineering*, vol. 13, no. 2, 2021.
- [4] Y. Bao, Z. Chen, S. Wei, Y. Xu, Z. Tang, and H. Li, "The state of the art of data science and engineering in structural health monitoring," *Engineering*, vol. 5, no. 2, pp. 234–242, 2019.
- [5] X. Zhu and F. L. di Scalea, "Thermal stress measurement in continuous welded rails using the hole-drilling method," *Experimental Mechanics*, vol. 57, no. 1, pp. 165–178, 2017.
- [6] C. Wang, X. Yu, M. Jiang, Z. Xing, and C. Wang, "Numerical and experimental investigation into the evolution and distribution of residual stress in laser transmission welding of PC/Cu/PC," *Optics & Laser Technology*, vol. 136, p. 106786, 2021.
- [7] J. Alegre, A. Díaz, I. Cuesta, and J. Manso, "Analysis of the influence of the thickness and the hole radius on the calibration coefficients in the hole-drilling method for the determination of non-uniform residual stresses," *Experimental Mechanics*, vol. 59, no. 1, pp. 79–94, 2019.
- [8] Y. Li, J. Wu, B. Qiang, S. Zhou, W. Liu, and C. Yao, "Measurements of residual stresses in a welded orthotropic steel deck by the hole-drilling method considering stress biaxiality," *Engineering Structures*, vol. 230, p. 111690, 2021.
- [9] C. Ruud, "A review of selected non-destructive methods for residual stress measurement," *NDT international*, vol. 15, no. 1, pp. 15–23, 1982.
- [10] G. Liu, H. Liu, A. Wei, J. Xiao, P. Wang, and S. Li, "A new device for stress monitoring in continuously welded rails using bi-directional strain method," *Journal of Modern Transportation*, vol. 26, no. 3, pp. 179–188, 2018.

- [11] Y. Ou, K. E. Tatsis, V. K. Dertimanis, M. D. Spiridonakos, and E. N. Chatzi, "Vibration-based monitoring of a small-scale wind turbine blade under varying climate conditions. part i: An experimental benchmark," *Structural Control and Health Monitoring*, vol. 28, no. 6, p. e2660, 2021.
- [12] N. Milosevic, B. Younise, A. Sedmak, M. Travica, and A. Mitrovic, "Evaluation of true stress–strain diagrams for welded joints by application of digital image correlation," *Engineering Failure Analysis*, vol. 128, p. 105609, 2021.
- [13] W. Wang, C. Xu, H. Jin, S. Meng, Y. Zhang, and W. Xie, "Measurement of high temperature full-field strain up to 2000 °c using digital image correlation," *Measurement Science and Technology*, vol. 28, no. 3, p. 035007, 2017.
- [14] W. Wang, C. Xu, Y. Zhang, Y. Zhou, S. Meng, and Y. Deng, "An improved ultrasonic method for plane stress measurement using critically refracted longitudinal waves," *NDT & E International*, vol. 99, pp. 117–122, 2018.
- [15] Z. Li, J. He, J. Teng, Q. Huang, and Y. Wang, "Absolute stress measurement of structural steel members with ultrasonic shear-wave spectral analysis method," *Structural Health Monitoring*, vol. 18, no. 1, pp. 216–231, 2019.
- [16] J. M. Hughes, J. Vidler, C.-T. Ng, A. Khanna, M. Mohabuth, L. F. Rose, and A. Kotousov, "Comparative evaluation of in situ stress monitoring with rayleigh waves," *Structural Health Monitoring*, vol. 18, no. 1, pp. 205–215, 2019.
- [17] H. Lee, H. J. Lim, T. Skinner, A. Chattopadhyay, and A. Hall, "Automated fatigue damage detection and classification technique for composite structures using lamb waves and deep autoencoder," *Mechanical Systems and Signal Processing*, vol. 163, p. 108148, 2022.
- [18] C. Villares Holguin, H. Hultmann Ayala, and A. Kubrusly, "Improved stress estimation with machine learning and ultrasonic guided waves," *Experimental Mechanics*, pp. 1–15, 2021.
- [19] V. V. Mishakin, S. Dixon, and M. D. G. Potter, "The use of wide band ultrasonic signals to estimate the stress condition of materials," *Journal of Physics D: Applied Physics*, vol. 39, pp. 4681–4687, oct 2006.
- [20] D. Hughes and J. Kelly, "Second-order elastic deformation of solids," *Physical Review*, vol. 92, no. 5, pp. 1145–1149, 1953.

- [21] Y. Pao, W. Sachse, and H. Fukuoka, "Acoustoelasticity and ultrasonic measurements of residual stresses," *Phys. Acoust.*, vol. 17, pp. 61–143, 1984.
- [22] J. L. Rose, *Ultrasonic Guided waves in solid media*. Cambridge University Press, 2014.
- [23] N. Gandhi, J. E. Michaels, and S. J. Lee, "Acoustoelastic lamb wave propagation in biaxially stressed plates," *The Journal of the Acoustical Society of America*, vol. 132, no. 3, pp. 1284–1293, 2012.
- [24] N. Pei and L. J. Bond, "Higher order acoustoelastic lamb wave propagation in stressed plates," *The Journal of the Acoustical Society of America*, vol. 140, no. 5, pp. 3834–3843, 2016.
- [25] A. C. Kubrusly, A. M. B. Braga, and J. P. von der Weid, "Derivation of acoustoelastic lamb wave dispersion curves in anisotropic plates at the initial and natural frames of reference," *The Journal of the Acoustical Society of America*, vol. 140, no. 4, pp. 2412–2417, 2016.
- [26] Y. Yang, C. T. Ng, M. Mohabuth, and A. Kotousov, "Finite element prediction of acoustoelastic effect associated with lamb wave propagation in pre-stressed plates," *Smart Materials and Structures*, vol. 28, p. 095007, jul 2019.
- [27] A. Pau and F. Lanza di Scalea, "Nonlinear guided wave propagation in prestressed plates," *The Journal of the Acoustical Society of America*, vol. 137, no. 3, pp. 1529–1540, 2015.
- [28] A. C. Kubrusly, N. Pérez, T. F. Oliveira, J. C. Adamowski, A. M. B. Braga, and J. P. Von de Weid, "Mechanical strain sensing by broadband time reversal in plates," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, pp. 746–756, 2016.
- [29] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.
- [30] R. Vieira and J. Lambros, "Machine learning neural-network predictions for grain-boundary strain accumulation in a polycrystalline metal," *Experimental Mechanics*, vol. 61, no. 4, pp. 627–639, 2021.
- [31] Z. Liu, Q. Peng, X. Li, C. He, and B. Wu, "Acoustic emission source localization with generalized regression neural network based on time difference

- mapping method," *Experimental Mechanics*, vol. 60, no. 5, pp. 679–694, 2020.
- [32] R. Wu, C. Kong, K. Li, and D. Zhang, "Real-time digital image correlation for dynamic strain measurement," *Experimental Mechanics*, vol. 56, no. 5, pp. 833–843, 2016.
- [33] X.-C. Zhang, J.-G. Gong, and F.-Z. Xuan, "A deep learning based life prediction method for components under creep, fatigue and creep-fatigue conditions," *International Journal of Fatigue*, vol. 148, p. 106236, 2021.
- [34] H. Lim and H. Sohn, "Online stress monitoring technique based on lamb-wave measurements and a convolutional neural network under static and dynamic loadings," *Experimental Mechanics*, vol. 60, no. 2, pp. 171–179, 2020.
- [35] Q. Ji, L. Jian-Bin, L. Fan-Rui, Z. Jian-Ting, and W. Xu, "Stress evaluation in seven-wire strands based on singular value feature of ultrasonic guided waves," *Structural Health Monitoring*, p. 14759217211005399, 2021.
- [36] M. Rautela and S. Gopalakrishnan, "Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks," *Expert Systems with Applications*, vol. 167, p. 114189, 2021.
- [37] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, 2020.
- [38] H. Chahal, H. Toner, and I. Rahkovsky, "Small data's big ai potential," 2021.
- [39] A. Entezami, H. Sarmadi, and B. Razavi, "An innovative hybrid strategy for structural health monitoring by modal flexibility and clustering methods," *Journal of Civil Structural Health Monitoring*, vol. 10, 11 2020.
- [40] M. Mohammad, A. Tajuddin, S. Abdullah, N. Jamaluddin, and B. Murat, "Fatigue damage monitoring using unsupervised clustering method of acoustic emission signal on sae 1045 steel," *International Journal of Automotive and Mechanical Engineering*, vol. 13, pp. 3584–3598, 12 2016.
- [41] Y. Sun, Y. Xu, W. Li, Q. Li, X. Ding, and W. Huang, "A lamb waves based ultrasonic system for the simultaneous data communication, defect inspection, and power transmission," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2021.

- [42] H. Chen, Z. Liu, Y. Gong, B. Wu, and C. He, "Evolutionary strategy-based location algorithm for high-resolution lamb wave defect detection with sparse array," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 68, no. 6, pp. 2277–2293, 2021.
- [43] M. Mitra and S. Gopalakrishnan, "Guided wave based structural health monitoring: A review," *Smart Materials and Structures*, vol. 25, p. 053001, mar 2016.
- [44] M. Harb and F.-G. Yuan, "Barely visible impact damage imaging using non-contact air-coupled transducer/laser doppler vibrometer system," *Structural Health Monitoring*, vol. 16, 11 2016.
- [45] M. N. Ahmed, "A study of guided ultrasonic wave propagation characteristics in thin aluminum plate for damage detection," Master's thesis, University of Toledo, may 2014.
- [46] R. DeNale, J. D. Achenbach, and Y. Rajapakse, *Solid mechanics research for quantitative non-destructive evaluation: Proceedings of the ONR Symposium on Solid Mechanics Research for QNDE, Northwestern University, Evanston, IL*. Springer Netherlands, September 1985.
- [47] Y. Ma, Z. Yang, J. Zhang, K. Liu, Z. Wu, and S. Ma, "Axial stress monitoring strategy in arbitrary cross-section based on acoustoelastic guided waves using pzt sensors," *AIP Advances*, vol. 9, no. 12, p. 125304, 2019.
- [48] F. Shi, J. E. Michaels, and S. J. Lee, "In situ estimation of applied biaxial loads with lamb waves," *The Journal of the Acoustical Society of America*, vol. 133, no. 2, pp. 677–687, 2013.
- [49] L. M. Martinho, A. C. Kubrusly, N. Pérez, and J. P. von der Weid, "Strain sensitivity enhancement of broadband ultrasonic signals in plates using spectral phase filtering," *Applied Sciences*, vol. 11, no. 6, 2021.
- [50] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2nd ed., 2019.
- [51] A. Subasi, *Practical Machine Learning for Data Analysis Using Python*. Academic Press, 1 ed., 2020.
- [52] A. H. Hoss Belyadi, *Machine Learning Guide for Oil and Gas Using Python: A Step-by-Step Breakdown with Data, Algorithms, Codes, and Applications*. Gulf Professional Publishing, 2021.

- [53] C. Bishop, *Pattern Recognition and Machine Learning*. Information science and statistics, New York, NY, USA: Springer, 2006.
- [54] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. Springer, 2 ed., 2008.
- [55] J.-P. Briot, G. Hadjeres, and F. Pachet, "Deep learning techniques for music generation - a survey," 09 2017.
- [56] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, p. 1–37, Dec. 2007.
- [57] M. M. Adankon and M. Cheriet, *Support Vector Machine*, pp. 1303–1308. Boston, MA: Springer US, 2009.
- [58] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [59] V. N. Vapnik, *Statistical Learning Theory*. New York, NY: Wiley, 1998.
- [60] M. Sabzekar and S. M. H. Hasheminejad, "Robust regression using support vector regressions," *Chaos, Solitons & Fractals*, vol. 144, p. 110738, 2021.
- [61] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. New York, USA: Springer, 2014.
- [62] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [63] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, USA: Wiley, 2 ed., 2000.
- [64] S. Zhang, "Challenges in knn classification," *IEEE Transactions on Knowledge & Data Engineering*, p. (in press), 2021.
- [65] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, (San Diego, CA, USA), 2015.
- [66] G. Strang, *Linear Algebra and Learning from Data*. Wellesley, USA: Wellesley-Cambridge Press, 2019.

- [67] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [68] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.
- [69] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [70] A. Jones, C. Kruger, and B. Johnston, *The Unsupervised Learning Workshop: Get started with unsupervised learning algorithms and simplify your unorganized data to help make future predictions*. Code. Packt Publishing, 2020.
- [71] T. Caliński and H. JA, "A dendrite method for cluster analysis," *Communications in Statistics—Theory and Methods*, vol. 3, pp. 1–27, 01 1974.
- [72] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE International Conference on Data Mining*, pp. 911–916, 2010.
- [73] B. J. D. Sitompul, O. S. Sitompul, and P. Sihombing, "Enhancement clustering evaluation result of davies-bouldin index with determining initial centroid of k-means algorithm," *Journal of Physics: Conference Series*, vol. 1235, p. 012015, jun 2019.
- [74] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
- [75] F. Gewers, G. Rodrigues Ferreira, H. Arruda, F. Silva, C. Comin, D. Amancio, and L. da F. Costa, "Principal component analysis: A natural approach to data exploration," *ACM Computing Surveys*, vol. 54, pp. 1–34, 05 2021.
- [76] S. Brunton and J. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. New York, NY, USA: Cambridge University Press, 2019.
- [77] L. van der Maaten and G. Hinton, "Vializing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 11 2008.
- [78] B. Johnston, A. Jones, and C. Kruger, *Applied Unsupervised Learning with Python*. Packt Publishing, 1st edition ed., 2019.

- [79] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 11 2008.
- [80] I. Gijbels and M. Hubert, “1.07 - robust and nonparametric statistical methods,” in *Comprehensive Chemometrics* (S. D. Brown, R. Tauler, and B. Walczak, eds.), pp. 189–211, Oxford: Elsevier, 2009.
- [81] P. Domingos, “A few useful things to know about machine learning,” *Communications of the ACM*, vol. 55, no. 10, p. 78–87, 2012.
- [82] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, USA: Springer, 2013.
- [83] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012.
- [84] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [85] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, D. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, and X. Zhang, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.
- [86] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [87] M. Allen, D. Poggiali, K. Whitaker, T. Marshall, J. Langen, and R. Kievit, “Raincloud plots: a multi-platform tool for robust data visualization,” *Wellcome Open Research*, vol. 4, p. 63, 01 2021.
- [88] K. Deb, “Multi-objective optimisation using evolutionary algorithms: an introduction,” in *Multi-objective evolutionary optimisation for product design and manufacturing*, pp. 3–34, Springer, 2001.
- [89] P. Ren, Y. Xiao, X. Chang, P.-y. Huang, Z. Li, X. Chen, and X. Wang, “A comprehensive survey of neural architecture search: Challenges and solutions,” *ACM Computing Surveys*, vol. 54, May 2021.

- [90] H. Hultmann Ayala, D. Muñoz, C. Llanos, and L. Coelho, "Efficient hardware implementation of radial basis function neural network with customized-precision floating-point operations," *Control Engineering Practice*, vol. 60, pp. 124–132, 03 2017.
- [91] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, N. Tran, and Z. Wu, "Fast inference of deep neural networks in FPGAs for particle physics," *Journal of Instrumentation*, vol. 13, pp. P07027–P07027, jul 2018.
- [92] A. Shawahna, S. Sait, and A. El-Maleh, "Fpga-based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7, pp. 7823–7859, 2019.
- [93] J. Zhu, L. Wang, H. Liu, S. Tian, Q. Deng, and J. Li, "An efficient task assignment framework to accelerate dpu-based convolutional neural network inference on fpgas," *IEEE Access*, vol. 8, pp. 83224–83237, 2020.
- [94] M. Siracusa and F. Ferrandi, "Tensor optimization for high-level synthesis design flows," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 4217–4228, 2020.
- [95] C.-H. Huang, "An fpga-based hardware/software design using binarized neural networks for agricultural applications: A case study," *IEEE Access*, vol. 9, pp. 26523–26531, 2021.
- [96] H. Bharadwaj, A. Agarwal, V. Chamola, R. Lakkaniga, V. Hassija, M. Guizani, and B. Sikdar, "A review on the role of machine learning in enabling iot based healthcare applications," *IEEE Access*, vol. 9, pp. 38859–38890, 2021.
- [97] K. Peddeti and S. Santhanam, "Dispersion curves for lamb wave propagation in prestressed plates using a semi-analytical finite element analysis," *The Journal of the Acoustical Society of America*, vol. 143, no. 2, pp. 829–840, 2018.
- [98] P. Zuo, X. Yu, and Z. Fan, "Acoustoelastic guided waves in waveguides with arbitrary prestress," *Journal of Sound and Vibration*, vol. 469, p. 115113, 2020.
- [99] E. Figueiredo, J. Figueiras, G. Park, C. R. Farrar, and K. Worden, "Influence of the autoregressive model order on damage detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 3, pp. 225–238, 2011.

- [100] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, "A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications," *Mechanical Systems and Signal Processing*, vol. 147, p. 107077, 2021.
- [101] D.-C. Feng, Z.-T. Liu, X.-D. Wang, Z.-M. Jiang, and S.-X. Liang, "Failure mode classification and bearing capacity prediction for reinforced concrete columns based on ensemble machine learning algorithm," *Advanced Engineering Informatics*, vol. 45, p. 101126, 2020.
- [102] D.-C. Feng, W.-J. Wang, S. Mangalathu, G. Hu, and T. Wu, "Implementing ensemble learning methods to predict the shear strength of rc deep beams with/without web reinforcements," *Engineering Structures*, vol. 235, p. 111979, 2021.
- [103] A. Blaiech, K. Ben Khalifa, C. Valderrama Sakuyama, M. Fernandes, and M. Bedoui, "A survey and taxonomy of fpga-based deep learning accelerators," *Journal of Systems Architecture*, vol. 98, pp. 331–345, 2019.
- [104] T. Kowsalya, "Area and power efficient pipelined hybrid merged adders for customized deep learning framework for fpga implementation," *Microprocessors and Microsystems*, vol. 72, p. 102906, 2020.
- [105] D. Danopoulos, C. Kachris, and D. Soudris, "Utilizing cloud fpgas towards the open neural network standard," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100520, 2021.
- [106] A. Aguiar, F. Neves Dos Santos, L. Santos, V. Filipe, and A. Sousa, "Vineyard trunk detection using deep learning – an experimental device benchmark," *Computers and Electronics in Agriculture*, vol. Volume 175, 06 2020.
- [107] H. Tabani, F. Mazzocchetti, P. Benedicte, J. Abella, and F. Cazorla, "Performance analysis and optimization opportunities for nvidia automotive gpus," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 21–32, 02 2021.
- [108] R. Giubilato, S. Chiodini, M. Pertile, and S. Debei, "An evaluation of ros-compatible stereo visual slam methods on a nvidia jetson tx2," *Measurement*, vol. 140, pp. 161–170, 2019.
- [109] S. Mittal, "A survey on optimized implementation of deep learning models on the nvidia jetson platform," *Journal of Systems Architecture*, vol. 97, pp. 428–442, 2019.