UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
DAPARTAMENTO DE ENGENHARIA DE ESTRUTURAS

Leonardo de Paula Silva Ferreira

CONTRIBUTIONS ON SHM IN COMPOSITE MATERIALS: LAMB WAVE ANALYSIS VIA 1-D-CNN WITH NUMERICAL MODELS

Belo Horizonte
2024

Leonardo de Paula Silva Ferreira

CONTRIBUTIONS ON SHM IN COMPOSITE MATERIALS: LAMB WAVE ANALYSIS VIA 1-D-CNN WITH NUMERICAL MODELS

Thesis submitted to the Programa de Pós-Graduação em Engenharia de Estruturas in partial fulfillment of the requirements for obtaining the title of Doctor in Structural Engineering.

Advisor: Prof. Dr. Carlos Alberto Cimini Jr.

Belo Horizonte

2024

F383c

Ferreira, Leonardo de Paula Silva.
Contributions on SHM in composite materials [recurso eletrônico] : lamb wave analysis via 1-D-CNN with numerical models / Leonardo de Paula Silva Ferreira. - 2024.
1 recurso online (189 f. : il., color.) : pdf.

Orientador: Carlos Alberto Cimini Jr.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 188-189.

Bibliografia: f. 172-187.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia de estruturas - Teses. 2. Materiais compostos - Teses. 3. Lamb, Ondas de - Teses. 4. Redes neurais convolucionais - Teses. 5. Inferência bayesiana - Teses. 6. Método dos elementos finitos - Teses. 7. Modelagem - Teses. I. Cimini Júnior, Carlos Alberto. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 624(043)

Ficha catalográfica elaborada pela Bibliotecária Roseli Alves de Oliveira CRB/6 2121
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ESTRUTURAS

PROPEEs

UFMG

ATA DA DEFESA DE TESE DE DOUTORADO EM ENGENHARIA DE ESTRUTURAS Nº: 98 DO ALUNO LEONARDO DE PAULA SILVA FERREIRA

Às **09:00** horas do dia **26** do mês de **janeiro** de **2024**, reuniu-se em ambiente virtual da Escola de Engenharia da Universidade Federal de Minas Gerais - UFMG, a Comissão Examinadora indicada pelo Colegiado do Programa em **24 de novembro de 2023**, para julgar a defesa da Tese de Doutorado intitulada **"Contributions on SHM in Composite Materials: Lamb Wave Analysis Via 1D-CNN with Numerical Models"**, cuja aprovação é um dos requisitos para a obtenção do Grau de DOUTOR EM ENGENHARIA DE ESTRUTURAS na área de ESTRUTURAS.

Abrindo a sessão, o Presidente da Comissão, **Prof. Dr. Carlos Alberto Cimini Jr**, após dar a conhecer aos presentes o teor das Normas Regulamentares passou a palavra ao candidato para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores, com a respectiva defesa do candidato. Logo após, a Comissão se reuniu, sem a presença do candidato e do público, para julgamento e expedição do resultado final.

Prof. Dr. Carlos Alberto Cimini Jr - DEES - UFMG (Orientador)
Prof. Dr. Lázaro Valentim Donadon - DEMEC - UFMG
Prof. Dr. Leonardo Antônio Borges Torres - DELT - UFMG
Prof. Dr. Samuel da Silva - UNESP - Ilha Solteira
Prof. Dr. Elói João Faria Figueiredo - Universidade Lusófona de Lisboa

Após reunião, a Comissão considerou o candidato APROVADO, conforme pareceres em anexo.

O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão.

Nada mais havendo a tratar, o Presidente encerrou a reunião e lavrou a presente ATA, que será assinada por todos os membros participantes da Comissão Examinadora.

Belo Horizonte, 26 de janeiro de 2024.

Observações:

1. A aprovação do candidato na defesa da Tese de Doutorado não significa que o mesmo tenha cumprido todos os requisitos necessários para obtenção do Grau de Doutor em Engenharia de Estruturas;
2. Este documento não terá validade sem a assinatura do Coordenador do Programa de Pós-Graduação.

Documento assinado eletronicamente por **Carlos Alberto Cimini Junior**, **Professor do Magistério Superior**, em 26/01/2024, às 13:30, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Leonardo Antonio Borges Torres**, **Professor do Magistério Superior**, em 26/01/2024, às 13:30, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Lazaro Valentim Donadon**, **Membro de comissão**, em 26/01/2024, às 13:31, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Samuel da Silva**, **Usuário Externo**, em 26/01/2024, às 13:31, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Elói João Faria Figueiredo**, **Usuário Externo**, em 26/01/2024, às 13:32, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Felicio Bruzzi Barros**, **Coordenador(a) de curso de pós-graduação**, em 15/02/2024, às 10:38, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2967786** e o código CRC **73F00384**.

For my wife, Danyella, who has shared in every challenge and triumph.

And to my parents, whose boundless love and sacrifices have shaped my world.

Acknowledgments

This work represents years of dedication, culminating in results I am deeply proud of. However, this journey wouldn't have been possible without the invaluable contributions of many, to whom I owe my deepest gratitude.

First and foremost, I thank God for the gift of life and the unending strength and guidance provided throughout this journey.

I am immensely grateful to my advisor, Professor Carlos Cimini Jr., for his guidance and unwavering confidence in my work throughout my Ph.D. journey. His support, understanding, and empathy towards his students are truly commendable. Professor Cimini Jr. has a unique ability to appreciate each individual's situation, always striving to offer help within his means. His dedication has been a cornerstone of my academic experience.

I am grateful to Professor Samuel Silva for his valuable advice during my doctorate and for my research intershing in Portugal through the CAPES/FCT partnership.

I extend my thanks to all the professors who have influenced my life. Your inspiration fueled my desire to become an educator and researcher. Although it's impossible to mention everyone without the risk of omission, I must extend a special thanks to one of my first teachers, Ms. Sônia, whose insight into my potential when I was just 8 years old set me on this path. Her advice to my parents to seek better educational opportunities for me and my brother was invaluable. I often ponder where life might have led without her guidance; it's hard to imagine that I would be pursuing a Ph.D. today without her early influence.

My heartfelt thanks go to my friends and professors in Lisbon, including Professor Eloi Figueiredo, Professor Dragos Ionut, Professor Adilia Charmier, Professor Nuno Maia, and friends Marcus Omori and Carol Altafim. Your warmth and camaraderie made Lisbon a second home and one of the most memorable chapters of my life. I will never forget all the *imperiais*, *pica-paus* and *caracois*.

A special thank to Profesor Rafael Teloli, initially a research partner from whom I learned a lot, but then became a friend along the way. Thank you for all your support

and friendship, which have been invaluable to me.

I acknowledge the Programa de Pós-Graduação em Engenharia de Estruturas (PROPEEs) and the Department of Structural Engineering (DEES) for their technical support and assistance whenever needed. I also thank the Department of Mechanical Engineering (DEMEC) and the Group of Acoustic and Vibrations (GRAVi), particularly Professor Lázaro Donadon, for their invaluable support during our experimental campaigns. Their assistance was especially critical during the pandemic when access to university facilities was significantly limited.

My sincere appreciation goes to the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES) for their financial support during my research internships at the Lisbon University and Lusófona University (Grant No. 88887.647575/2021-00).

To the Edson and Eni Castro family: thank you for being a second family to me. Much of my character and values were shaped by your influence, and for that, I am eternally grateful.

I extend my gratitude to my entire family, especially uncle Paulo Sergio and aunt Mauriceia Vieira, for being pillars in engineering and academia. Tio Paulo inspired me to be a Mechanical engineer and gave me my first scientific calculator, and tia Ceia always encouraged me to go further in my a higher education.

To my brothers, Lucas and Guilherme: thank you for the countless life experiences, laughter, and unwavering partnership. Love you both deeply.

A special thank you to my parents, José Leonardo (*in memoriam*) and Laurici. I would never have become who I am without your love, guidance, and support. Every educational opportunity and accomplishment I have achieved can be traced back to their sacrifices. They gave up so much to ensure that their sons had the best education possible, and for that, I am eternally grateful.

Finally and most importantly, I extend my deepest gratitude to my wife, Danyella, the love of my life. Her unwavering support and companionship have been the bedrock of this journey. Her sacrifices, particularly in understanding and supporting our separation due to my move to Portugal so soon after our marriage, have been crucial to my success. Her presence and encouragement have been my greatest source of strength and motivation.

To all of you, my heartfelt thanks.

"seule l'éducation est apte à sauve nos sociétés d'une dissolution possible, violente ou graduelle"

Jean Piaget (⋆1896, †1980).

Abstract

Lamb waves are a promising technique for structural health monitoring (SHM) of thin-walled structures due to their sensitivity to various types of damage and their ability to cover large areas with minimal sensor deployment. The main objective of this Thesis is to propose new contributions in the application of 1D Convolutional Neural Networks (CNNs) in damage detection on composite structures using Lamb waves, focusing mainly on evaluating the impact of experimental data availability and proposing ways to reduce experimental data limitations related to training machine learning algorithms. The first part of the study examines the sensitivity of 1D CNNs to the volume of training data and damage localization accuracy, using these algorithms to analyze minimal pre-processed Lamb wave signals and compares them with traditional neural Multi-Layer Perceptrons (MLPs) networks fed with damage indices obtained from the signals. In the second part, a novel stochastic updating framework for FE models simulating Lamb waves based on Bayesian inference is proposed to incorporate experimental uncertainty into numerical simulations. The stochastic updating procedure is divided into two steps: a sensitivity analysis using Sobol Indices and a Bayesian inference process using Markov-chain Monte Carlo (MCMC) simulations and the Metropolis-Hastings sampling algorithm. Additionally, a novel surrogate model based on neural networks is introduced to substitute the FE model in the Bayesian inference process. Finally, in the third part, the study evaluates the performance of 1D CNNs using a Bayesian data-driven training approach, focusing on scenarios with constrained experimental data availability and varying temperatures. Comparative analyses are made using experimental, numerical, and hybrid datasets during the training phase. The study demonstrates the enhanced adaptability and robustness of the machine learning model when trained with a combination of experimental and numerically generated data. Key contributions of this thesis include (i) developing a minimal pre-processing 1D CNN strategy for SHM, (ii) a framework for stochastic FE model updating considering experimental uncertainties and temperature effects, (iii) the proposition of a neural network surrogate model to accelerate the Bayesian inference process on Lamb wave applications, and (iv) a methodology for training 1D CNNs with a limited experimental dataset.

Keywords: Structural health monitoring; Composite materials; Lamb waves; Convolutional neural networks; Bayesian inference; Finite element models; Surrogate modeling; Sobol Indices.

Resumo

As ondas Lamb são uma técnica promissora para monitoramento da saúde estrutural (SHM) de estruturas de paredes finas devido à sua sensibilidade a vários tipos de danos. O principal objetivo desta Tese é propor novas contribuições na aplicação de redes neurais convolucionais unidimensionais (1D CNNs) na detecção de danos em estruturas compostas usando ondas Lamb, focando principalmente na avaliação do impacto da disponibilidade de dados experimentais e propondo maneiras de reduzir as limitações de dados experimentais relacionadas ao treinamento de algoritmos de aprendizado de máquina. A primeira parte do estudo examina a sensibilidade das 1D CNNs ao volume de dados de treinamento e a precisão na localização de danos, utilizando esses algoritmos para analisar sinais de ondas Lamb e comparando os resultados com os obtidos por redes neurais tradicionais (MLPs) alimentadas com índices de danos calculados a partir dos sinais. Na segunda parte, é proposto um método de atualização estocástico baseado em inferencia Bayesiana para modelos de elementos finitos, com o intuito de incorporar incerteza experimental em simulações numéricas. O método de atualização estocástica é dividido em duas etapas: uma análise de sensibilidade usando Índices de Sobol e um processo de inferência Bayesiana usando simulações de Cadeia de Markov Monte Carlo (MCMC) e o algoritmo de amostragem de Metropolis-Hastings. Além disso, é proposto um novo modelo de substituição baseado em redes neurais para substituir o modelo de elementos finitos no processo de inferência Bayesiana. Finalmente, na terceira parte, o estudo avalia o desempenho das 1D CNNs em cenários com disponibilidade limitada de dados experimentais e variação de temperaturas. Análises comparativas são feitas usando conjuntos de dados experimentais, numéricos gerados por um modelo estocástico e híbridos durante a fase de treinamento. As principais contribuições desta tese incluem (i) o desenvolvimento de uma estratégia de localização de danos utilizando 1D CNNs, (ii) um método para atualização estocástica de modelos de elementos finitos considerando incertezas experimentais e efeitos de temperatura, (iii) a proposição de um modelo de substituição baseado em rede neurais para acelerar o processo de inferência Bayesiana em aplicações de ondas Lamb, e (iv) uma metodologia para treinar 1D CNNs com um conjunto de dados experimentais limitado.

Palavras-chave: Monitoramento da saúde estrutural; Materiais compostos; Ondas de Lamb; Redes neurais convolucionais; Inferência Bayesiana; Modelos de elementos finitos; Modelagem substituta; Índices de Sobol.

List of Figures

List of Tables

List of Acronyms

Adam	-	Adaptive moment estimation
AI	-	Artificial inteligence
AMP	-	Amplitude
ANN	-	Artificial neural network
AR	-	Autoregresive
BSS	-	Baseline signal stretch
CFRP	-	Carbon fiber reinforced plastic
CNN	-	Convolutional neural network
CONCEPT	-	CarbON-epoxi CompositE PlaTE
CONV	-	Convolutional layer
CWT	-	Continuous wavelet transform
DI	-	Damage index
DWT	-	Discrete wavelet transform
EXP	-	Experimental dataset
FC	-	Fully connected layer
FE	-	Finite element
FD	-	Finite differences
FFT	-	Fast Fourier transform
FRF	-	Frequency response function
GA	-	Genetic algorithm
GFRP	-	Glass fiber reinforced plastic
GP	-	Gaussian process
GRAVi	-	GRupo de Acústica e Vibrações (Acoustic and vibrations group)
GRU	-	Gated recurrent unit
HYB	-	Hybrid dataset
LaWaDe	-	Lamb wave detection
LISA	-	Linear interaction simulation approach
LLM	-	Large language model
MCMC	-	Markov chain-Monte Carlo

MLP	-	Multilayer perceprton
MTF	-	Markov transition field
MSE	-	Mean squared error
NASA	-	National Aeronautics and Space Administration
NDI	-	Non-destructive inspection
NUM	-	Numerical dataset
OBS	-	Optimal baseline subtraction
PCE	-	Polynomial chaos expansion
PDF	-	Probability density function
POOL	-	Pooling layer
PSD	-	Power spectral density
PZT	-	Lead-Zirconium-Titanium (Pb-Zi-Ti)
ReLU	-	Rectified linear unit
RGB	-	Red, green, and blue
RMS	-	Root mean square
RMSE	-	Root mean square error
STFT	-	Short-time Fourier transform
SGD	-	Stochastic gradient descent
SGDM	-	Stochastic gradient descent with momentum
SHM	-	Structural health monitoring
SVM	-	Support vector machine
TOF	-	Time of flight
TMCMC	-	Transitional Markov-chain Monte Carlo
TVDI	-	Time varying damage index
UFMG	-	Universidade Federal de Minas Gerais
WFE	-	Wave finite element

List of Symbols

Latin Symbols

$\hat{\boldsymbol{y}}$	Vector with preditions of the neural network
$\boldsymbol{\mathcal{N}}$	Neural network
$\boldsymbol{\mathcal{T}}$	Activation function
$\boldsymbol{X}$	Features
$\boldsymbol{x}_{i,j}$	Input of the j-th neuron of the i-th layer
$\boldsymbol{Y}$	Labels
$\mathbf{Y}_{target}$	Target values
$\boldsymbol{x}^{(l)}$	l-th sample feature in a dataset
$\boldsymbol{X}^{(Te)}$	Subset of test features
$\boldsymbol{X}^{(Tr)}$	Subset of training features
$\boldsymbol{X}^{(Val)}$	Subset of validation features
$\boldsymbol{y}^{(l)}$	l-th sample label in a dataset
$\boldsymbol{Y}^{(Te)}$	Subset of test labels
$\boldsymbol{Y}^{(Tr)}$	Subset of training labels
$\boldsymbol{Y}^{(Val)}$	Subset of validation labels
$\hat{y}_i$	Result for the i-th neuron on the output layer
$\mathcal{D}_{\mathcal{M}}$	Pertubed model predictions
$\mathcal{F}$	Mapping function
$\mathcal{L}(\cdot)$	Loss function

$\mathcal{M}$	Model
$\mathcal{M}^{PC}$	PCE model
$\mathcal{M}_0$	Expected mean
$\mathcal{M}_{ij}(\theta_i, \theta_j)$	Conditional mean for parameter ij $(i \neq j)$
$\mathcal{M}_i(\theta_i)$	Conditional mean for parameter i
$\mathcal{R}$	Regularization operator
b_{ij}	Bias term of the j-th neuron of the i-th layer
C	Convolution layer
c_L	Longitudinal wave velocity
c_T	Transversal wave velocity
D	Dropout layer
E	ReLU layer
E_1	Young's modulus in direction 1
E_2	Young's modulus in direction 2
FC	Fully-connected layer
G_{12}	In plane shear modulus
$G_{13}\text{and}G_{23}$	Out of plane shear modulus
H	Pooling filter size
h	Half thickness of the plate
I	Input layer
k	Wave number; number of selected channels
M	Number of points in an array
MSE	Mean squared error
N	Number of samples in a dataset

O	Output layer
P	Pooling layer
S	Stride
S_{ij}	Second-order sobol index
S_i	First-order sobol index
$Var[\cdot]$	Variance
W	Convolution kernel
w_i	Weight values
Z	Activated feature map
$w_{i,k,j}$	Weight that connects the j-th neuron of the i-th layer to the k-th neuron of the next layer
A0, A1, A2...	Antisymmetric wave modes
S0, S1, S2...	Symmetric wave modes

Greek Symbols

α	Learning rate; Metropolis-Hastings acceptance criteria
$\boldsymbol{\theta}$	Model parameters
$\boldsymbol{\theta}_0$	Taylor expansion central point
$\boldsymbol{\Theta}_{NN}$	Neural network set of parameters
$\Delta\theta_i$	Small variation of i-th parameter
ϵ	Error metric; Gaussian error
λ	Wavelength; regularization parameter
$\mathscr{D}$	Vector of measurements or observations
ν_{12}	In plane Poisson coeficient
ω	Circlar frequency $[rad/s]$
$\pi(\boldsymbol{\theta})$	Prior distribution

$\pi(\mathscr{D})$	Evidence or marginal likelihood
$\Psi_\alpha(\boldsymbol{\theta})$	Multivariate orthonormal polynomials
ρ	Mass density
σ_p	Standard deviation of the proposed candidates distributions
y_α	PCE coeficients

Number Sets

$\mathbb{R}^{1\times M}$	Onedimentional sample subspace
$\mathbb{R}^{1\times P}$	P-dimentional label subspace

Contents

1 INTRODUCTION

This chapter first introduces the structural health monitoring (SHM) approach, with a particular focus on the use of ultrasonic waves. The discussion then shifts to the motivation of the thesis, highlighting the main challenges associated with this type of inspection technique and recent strategies employing machine learning algorithms. Subsequently, the scope of the thesis delineates all the steps that are covered throughout this work. Objectives and main contributions of the thesis are also articulated, and the chapter concludes with an outline of the work.

1.1 CONTEXTUALIZATION

Monitoring performance, conducting damage prognosis, and assessing the aging of structures hold critical significance in the field of engineering. As structures age, they become increasingly susceptible to damage, leading to a decline in their reliability (YUAN, 2016). The consequences of structural damage are severe, often resulting in substantial financial losses and eventually loss of lifes. For example, data from the US National Transportation Safety Board including all types of aircraft incidents reveals that between 2000 and 2019, there were 29,093 aircraft accidents globally, encompassing both general and commercial aviation airplanes. Out of these incidents, 6,376 (21.9%) were fatal, leading to 17,907 fatalities. In Brazil alone, there were 185 accidents during this period, with 155 being fatal and resulting in 574 fatalities (National Transportation Safety Board, 2023).

The process of structural monitoring began with non-destructive inspection (NDI) and its variants. A systematic NDI approach, conducted at regular intervals, aims to detect structural damage or material degradation during routine inspections. This is done to avert potential structural failures that could otherwise lead to more severe and disastrous outcomes. Traditional NDI techniques include visual inspection, ultrasonic scanning, radioscopy, dye penetrant testing, shearography, magnetic resonance imagery, laser interferometry, infrared thermography, eddy current, etc. (ACHENBACH, 2000).

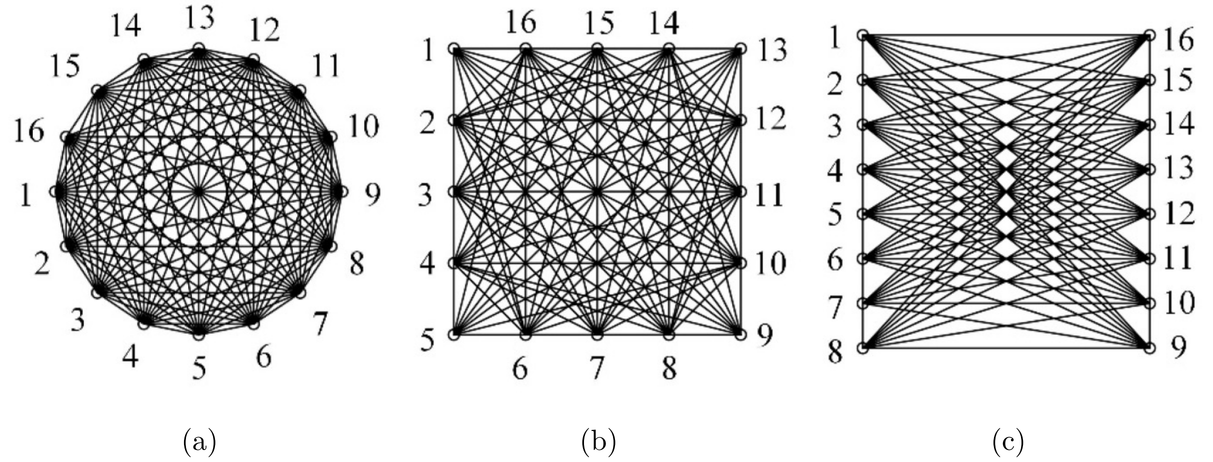
While crucial in preventing structural failures, many NDI-based inspections are carried out periodically, often overlooking changes in operating conditions or ongoing structural deterioration. For instance, Yuan (2016) highlights two examples to illustrate the limitations of NDI: the detachment of the vertical stabilizer from Flight TS 961 (Airbus A310) in 2005 and the grounding of 128 Boeing 737 aircraft in the Southwest Airlines fleet in 2015. The former incident occurred just five days after a routine check, with the next major inspection scheduled for the following year. The latter incident was due to missed mandatory inspections, leading to approximately 80 flight cancellations in a single day. Consequently, traditional inspection methods encounter challenges in NDI, particularly in achieving automatic and real-time assessments, owing to their point-by-point approach and reliance on bulky devices suited for offline, periodic maintenance.

In recent decades, the application of composite materials has increased across various industries, including aerospace, civil engineering, and energy generation. Particularly in the aerospace industry, the utilization of composite materials, such as carbon fiber reinforced plastics (CFRP) and glass fiber reinforced plastics (GFRP) is a reality for commercial aircraft. However, composite structures are susceptible to multiple failure modes, including fiber damage, matrix cracking, delamination between different plies, fiber debonding, and shear-driven fracture (ZHANG *et al.*, 2020). This vulnerability presents additional challenges for current NDI techniques, necessitating the development of more advanced and sensitive methods for detecting and assessing damage in composite materials.

Enabled by technological advances and breakthroughs in sensor technology, some NDI methods have evolved into SHM (MITRA; GOPALAKRISHNAN, 2016). The primary distinction between SHM and NDI lies in their operational timing: SHM aims for systems that operate in real-time, whereas NDI involves systems used during scheduled maintenance stops. The fundamental objective of SHM is to identify changes at the earliest possible opportunity, enabling the scheduling of corrective action to minimize downtime, operational costs, and maintenance expenses, while also reducing the risk of catastrophic failures during operation.

Various types of SHM systems have been studied in recent decades, among which those based on vibrations (LI *et al.*, 2022), eddy current (HA; LEE; LEE, 2021), and piezoelectric (PARK *et al.*, 2006) sensors can be highlighted. One of the main candidates are techniques based on ultrasonic guided waves, which have been especially developed in the last two decades (IHN; CHANG, 2008; MITRA; GOPALAKRISHNAN, 2016; YANG *et al.*, 2023). The main advantages of guided wave-based techniques are: (i) the involved transducers are inexpensive and can be easily incorporated into the production process of structures, (ii) they have the ability to inspect large areas with a relatively small number of sensors, (iii) they involve high excitation frequencies, which enables the detection of small flaws, (iv) low-frequency environmental components do not interfere with the techniques, and (v) the damping effects of the structure are not very prominent due to the transient nature of the method. The main problems are linked to the complexities associated with wave propagation and their interaction with damage.

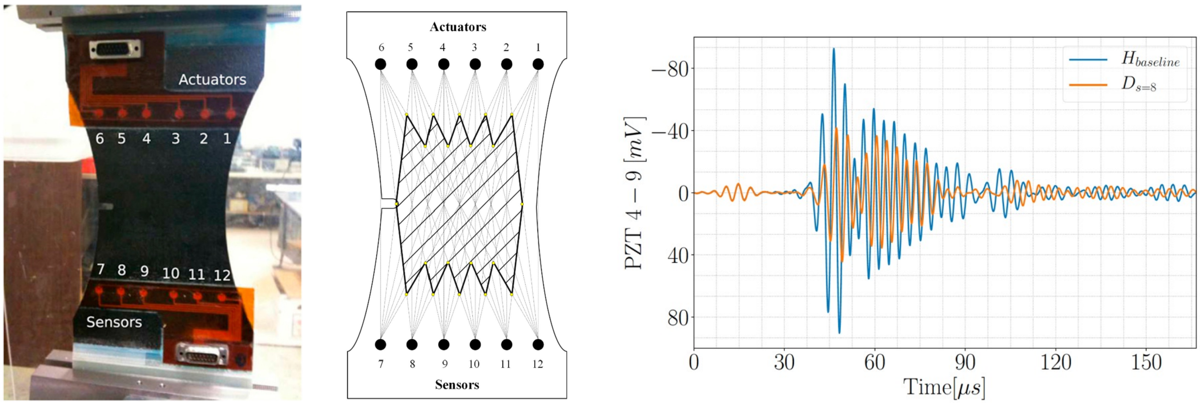
Among the various forms of guided waves, Lamb waves are utilized when dealing with thin structures with free boundaries. Due to their applications in thin-walled structures, Lamb wave inspections are versatile and can be applied to several industries, including aeronautical (De LUCA *et al.*, 2020), wind turbines (KHAZAEE; DERIAN; MOURAUD, 2022), and piping and pressure vessels (CARRINO; MAFFEZZOLI; SCARSELLI, 2021). For use in SHM, Lamb waves can be generated and captured in the structure through Lead-Zirconium-Titanium - Pb-Zi-Ti (PZT) - sensors (GIURGIUTIU, 2014). These sensors are attached to the structure in an array pattern and can inspect it on demand. Figures 1a, 1b and 1c describe circular, square and rectangular arrays, respectively.



Source: Adapted from Wang *et al.* (2020).

Figure 1 – Commom PZT arrays found on literature: (a) circular; (b) square; and (c) rectangular.

PZT sensors deform when subjected to electrical voltage and induce an acoustic vibration in the structure. This vibration propagates and can be measured by other sensors scattered throughout it. By measuring the behavior of Lamb waves in an undamaged structure, a ‘‘signature’’ (also called Baseline) can be obtained. Subsequent measurements are compared with the signature, and detection techniques can be used to locate and quantify the damages. Figure 2a presents a NASA benchmark dataset (SAXENA *et al.*, 2015) that uses Lamb wave to evaluate progressive damage on a CFRP coupon, Fig. 2b depicts the covered area by de combined sensor mesh, and Fig. 2c compares wave packages without and with damage to the structure.



(a) (b) (c)

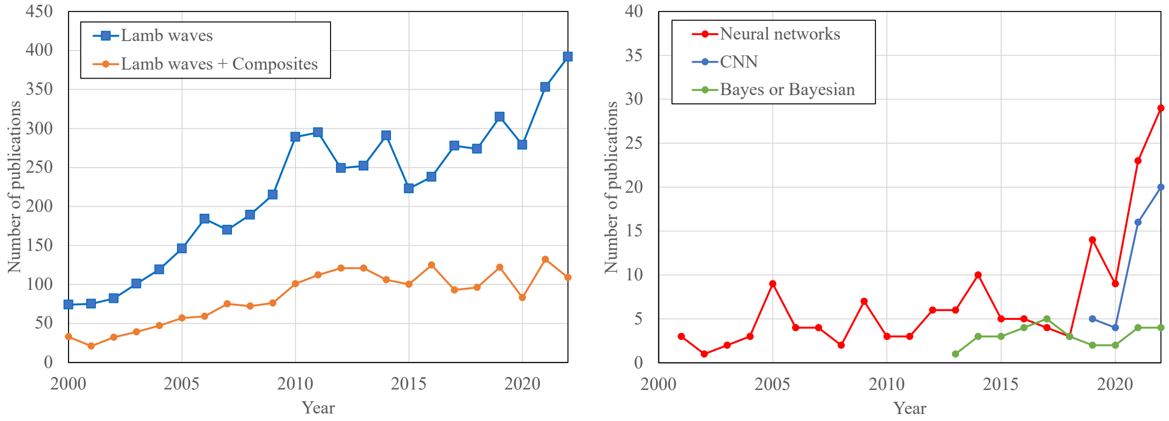
Source: Adapted from (a) and (c) Paixao *et al.* (2021), and (b) Wu *et al.* (2021).

Figure 2 – Example of experimental dataset made available by Saxena *et al.* (2015) (a) experiemental setup; (b) sensor layout; and (c) measured signals without and with damage between sensors one and nine.

Figure 2c illustrates the complex influence of damage on Lamb wave signals. The received signal exhibits delay and attenuation. Moreover, the presence of noise, environmental variations, multiple reflections due to damage and/or plate borders, and the multimodal nature of Lamb waves compound the difficulty of signal processing.

There has been substantial work in the signal processing of Lamb wave responses. Various techniques employ time-frequency analysis to isolate wave packets, identify wave modes, and assess changes in time of flight (TOF) (STASZEWSKI, 2002). These methods can implement multiple transformations, including wavelet transform (FENG; RIBEIRO; RAMOS, 2018), short-time Fourier transform (STFT) (KE *et al.*, 2019), Hilbert transform (SU *et al.*, 2019a), among others. However, time-frequency techniques often depend on isolating wave packets, a task that becomes challenging in scenarios with higher-order modes or multiple reflections. This complexity underscores the need for advanced signal processing strategies to accurately interpret Lamb wave signals, particularly in complex damage scenarios.

Other traditional damage localization methods are based on a damage index (DI) that represents changes in the state of the structure (ZHAO *et al.*, 2007; MUSTAPHA *et al.*, 2016). Generally, the DI should be designed to be sensitive to the presence of damage, and the accuracy of damage localization depends on the validity of the DI. However,



(a) (b)

Source: Prepared by the author.

Figure 3 – Number of publications in the last 23 years, according to SCOPUS, involving keyworks: (a) Lamb waves and Lamb waves + composites; (b) Lamb waves with neural networks, convolutional neural networks (CNN) or Bayes/Bayesian.

underlying but non-negligible influences, including temperature variations and changes in boundary conditions, can be sufficient to mask the damage-related features in signals and further weaken decision-making (YAN *et al.*, 2005; CROXFORD *et al.*, 2007; LIU *et al.*, 2015; SALMANPOUR; KHODAEI; ALIABADI, 2017).

In the last two decades, machine learning techniques, including Support Vector Machines (SVM) (ISA; RAJKUMAR, 2009; MECHBAL; URIBE; REBILLAT, 2015), Gaussian processes (SILVA *et al.*, 2020), and Artificial Neural Networks (ANN) (LIU *et al.*, 2013; DWORAKOWSKI *et al.*, 2015), have been employed in the processing of Lamb wave signals. Furthermore, deep-learning-based approaches (AZUARA; RUIZ; BARRERA, 2021; WU *et al.*, 2021; ZHANG *et al.*, 2022) utilize the high-dimensional feature mapping capabilities intrinsic to deep-learning algorithms. This enables the establishment of a relationship between Lamb wave signals and damage information, potentially enhancing the accuracy and efficiency of damage detection in SHM.

Figure 3 showcases a comprehensive survey conducted on the Scopus platform. It is evident from the data that the number of publications involving Lamb waves has been on a steady incline over the last two decades, culminating in an all-time high of 392 publications as of November 2023, as depicted in Fig. 3a. Research that use Lamb waves for composite materials inspection are also relevant, with an average of more than 100 studies per year in the last decade.

The data reveals a noteworthy trend in the field of Lamb waves and neural networks (Fig. 3b). Prior to 2019, the average number of publications per year in this domain hovered around 4.4 articles. However, post-2019, there has been a significant surge in the usage of the term, resulting in the publication of 110 articles between 2019 and 2022. Out of these 110 articles, 69 focus on convolutional neural networks (CNNs), highlighting the prominence of this type of architecture in the area. The earliest work combining CNNs with Lamb waves dates back to 2019. By 2023, this theme accounts for 20 out of the 29 articles published, underlining its growing importance and adoption in the field. Additionally, the number of publications simultaneously addressing Lamb waves and Bayes or Bayesian inference is notably low, with fewer than 5 works per year. This trend highlights a potentially underexplored area within the field, suggesting an opportunity for further investigation and development on the uncertainty quantification strategies on Lamb wave analysis.

1.2 MOTIVATION

Over the past decade, there has been a surge in application of deep-learing algorithms across various fields, ranging from image processing (MAHADEVKAR *et al.*, 2022) to large language models (LLMs) and generative AIs (HADI *et al.*, 2023). The area of SHM using Lamb waves also experienced this trend, and the scientific community proposed several new algorithms and applications using deep-learning. Section 2.3.3 showcases details of this evolution, and Yang *et al.* (2023) present a comprehensive literature review about machine learning for ultrasonic waves analysis.

Among promising algorithms, CNNs stand out due to their natural ability for feature extraction and classification within the same architecture (KIRANYAZ *et al.*, 2021). These algorithms are already state-of-the-art in some fields, as image recognition and video processing, and recent works have evaluated the use of classical two-dimensional CNNs for analyzing Lamb wave signals and detecting damage (LIU; ZHANG, 2019a; LOMAZZI *et al.*, 2023). In this architecture, the input data is a tensor with at least two dimensions, height and width, and an optional depth dimension. This approach is common in image processing, where each depth layer, also known as a channel, represents color information (for instance, red, green, and blue - RGB). This approach has been adapted to Lamb wave analysis, with the use of transformation algorithms that represent the wave propagation signal in some form of image to be classified by a CNN (SU *et al.*, 2019b; WU *et al.*, 2021). However, this method has the drawback of requiring some kind of transformation between the Lamb wave signal and the input image, which can be computationally expensive and/or may result in information loss. Very recently, one-dimensional convolutional neural networks (1D CNNs) have emerged as a promising research area for Lamb wave analysis (RAI; MITRA, 2022), as these networks can work directly with vector inputs, such as a time-domain signal representing a Lamb wave. Section 2.3.3 presents an extensive discussion about works using 2D CNNs and the recent proposals of 1D CNNs in Lamb wave analysis and other SHM applications. This work explores 1D CNNs for damage detection, localization and quantification, and provides guidelines for best practices in implementing these algorithms.

Furthermore, machine learning algorithms, especially neural networks, require substantial amount of data for training. Generating such a volume of data solely through experimental campaigns is not feasible for many applications. It is not cost-effective to test all possible damage scenarios, nor it is feasible to introduce certain types of damage into a structure simply to measure its response. Consequently, the use of numerical models to simulate scenarios and generate training data becomes integral to SHM. For these numerical models to be reliable, they must be calibrated with experimental data. Numerous studies have proposed training machine learning models to interpret Lamb wave signals using data generated from numerical simulations (SU; YE, 2005a; LU *et al.*, 2009; SBARUFATTI; MANSON; WORDEN, 2014), indicating a demand for reliable numerical models. Recent works, for instance, try to account for experimental uncertainty by introducing noise into the numerical results, often using random or Gaussian noise (RAI; MITRA, 2021). However, there has not yet been a study where experimental uncertainty is incorporated directly into the numerical model from experimental results. The author sees Bayesian inference as a promissing path to incorporate experimental uncertainty into finite element (FE) models that generate training datasets. These datasets would be able to include the experimental dispersion and help to train better machine learning algorithms. This is a subject of investigation addressed in this work.

Additionally, there is a lack of research in the literature that addresses the influence of temperature on Lamb wave simulations. This influence is usually included at material level through material tests in low and high temperatures, with property interpolation between them (MARIANI *et al.*, 2021). This work proposes an inverse approach that uses Lamb wave signals to estimate probability density function (PDF) for material properties and obtain a stochastic FE model that can be simulated within confidence intervals.

1.3 SCOPE OF THE THESIS

Having briefly addressed the main objectives and motivations behind SHM and the recent developments in the application of machine learning algorithms for Lamb wave signal analysis, the thesis delves into three pertinent topics concerning these algorithms: (i) the application of recent 1D CNN techniques to process Lamb wave signals in composite materials, exploring this promising and relatively new field, and the potential of these algorithms to enhance accuracy and efficiency of damage detection systems; (ii) the development of a reliable data generation strategy to complement experimental data with numerically generated data, aiming to provide a comprehensive dataset for training machine learning models; (iii) the evaluation of CNNs sensitivity to the type and amount of data used in the training stage, assessing the performance of the algorithm in scenarios with limited experimental data and temperature variations. Therefore, the thesis is organized into the three main parts presented below.

The first part of the thesis focuses on the development of an experimental system for the collection of Lamb wave signals. The Acoustics and Vibrations Group (GRAVi) of the mechanical engineering department at UFMG has a Lamb wave detection and analysis system called LaWaDe (Lamb Wave Detection System). LaWaDe consists of a signal acquisition platform for collecting Lamb waves signals integrated with post-processing software developed entirely within the research group. The signal acquisition platform is build using hardware and software from National Instruments®, i.e., the PXI® modular system automated with LabView® routines (BITTER; MOHIUDDIN; NAWROCKI, 2006). The post-processing routines are implemented using MATLAB®. In this part of the thesis, LaWaDe is enhanced, both in its physical aspect, with the inclusion of a signal switching platform that enables the automatic capture of multiple signals in a pitch-catch configuration, and modifications in its LabView interface to adapt the system to this new automation. Subsequently, the automated version of the system is evaluated in two tests for damage detection in a composite material structure using 1D CNNs: (i) a global position localization, and (ii) a local position analysis. Both tests are evaluated using a multilayer perceprtron (MLP) neural network approach fed by damage indices and a new proposal of 1D CNN fed directly with Lamb wave signals, without a pre-processing stage for damage index calculation. This part of the thesis aims to assess the capacity of the 1D CNNs and their sensitivity to the amount of data used in the training phase for damage localization.

The second part of the thesis introduces a stochastic updating framework for FE models, employing verification and validation steps. The core idea is to develop a stochastic model that incorporates experimental uncertainty and can supplement the experimental data with numerically generated data for the training step of machine learning algorithms. This model potentially reduces the number of experiments needed to create a machine learning classifier. The proposed framework is based in Bayesian inference and is structured in a three-step model update strategy: (i) initially, a deterministic analysis is conducted to determine optimal values for unknown properties, (ii) a sensitivity analysis using Sobol indices is performed to minimize the number of system parameters in the Bayesian inference stage, and (iii) a stochastic adjustment is applied using Markov-Chain Monte Carlo Simulations (MCMC) with the Metropolis-Hastings sampling algorithm to assign probability distribution functions (PDF) to relevant properties.

To reduce the computational time required for the MCMC process, the thesis proposes the use of a surrogate model based on ANNs, which offers faster computation compared to traditional FE models. The ANN model can be trained using parallelized Monte Carlo simulations, which contrasts with the sequential nature of the MCMC process, thereby further accelerating the updating procedure. This approach significantly reduces the time required for model updating without compromising the accuracy of the resulting probability density functions for uncertain parameters. The proposed framework for model updating and the surrogate modeling technique are evaluated using the CarbON-epoxi CompositE PlaTE (CONCEPT) database, provided by Silva (2018). Additionally, the framework is assessed under varying temperature conditions, aiming to compensate for temperature effects and incorporate this influence directly into the FE model, without prior knowledge of material properties.

Finally, the third part of the thesis employs the stochastic FE model to evaluate a Bayesian data-driven strategy for training 1D CNNs in different scenarios of experimental and numerical data availability. The performance of the classifiers is assessed based on the training strategies and the influence of temperature. The model updating strategy is evaluated for its effectiveness in generating numerical data at multiple temperatures and damage severities for training the classifier. This approach aims to enhance the robustness and adaptability of the classifier, ensuring its effectiveness across a range of conditions and data availability scenarios.

1.4 OBJECTIVES

The main objective of this doctoral thesis is to propose new contributions in the application of 1D CNNs in damage detection using Lamb waves, focusing mainly on evaluating the impact of experimental data availability and proposing ways to reduce experimental data limitations related to training machine learning algorihms. Furthermore, this work evaluates the performance of 1D CNNs on evaluating Lamb wave signals under varying temperature. In summary:

- to evaluate the performance of 1D CNNs when compared to classical MLP networks in tasks of detection and localization of damage in composite strutuctures using Lamb waves;
- to develop a framework for numerical data generation with the addition of experimental uncertainty and temperature effects into a FE model;
- to investigate performance of 1D CNNs in scenarios of limited experimental data, evaluating the impact of different strategies for training and the effect of temperature variation on model's accuracy;

1.5 MAIN CONTRIBUTIONS OF THE THESIS

The main contributions of this thesis to the literature are:

- The development of a 1D CNN strategy for evaluating Lamb wave signals with minimal pre-processing in SHM applications. The 1D CNNs are evaluated against classical MLP networks for damage localization under various scenarios of experimental data availability. Additionally, guidelines for training and utilizing these algorithms are provided;
- The development of a novel framework for stochastic updating of an FE model for a composite plate, taking into account experimental uncertainty and the influence of temperature on Lamb wave propagation. This updating approach, based on Bayesian inference, accommodates the uncertainty in Lamb wave data and the temperature effects on the measurement process, integrating them into the FE model. The updated FE model includes a confidence interval within which it can replicate the experimental setup accurately;

- The introduction of a surrogate modeling technique based on neural networks to address the computational challenges associated with Bayesian inference using MCMC simulations. This integration of Bayesian inference and machine learning in model updating can significantly reduce the computational cost associated with MCMC iterations;
- The proposal and evaluation of a Bayesian data-driven machine learning strategy for training 1D CNN models with constrained experimental data. This method employs Bayesian inference to update an FE model when experimental data is scarce, augmenting the dataset with synthesized Lamb wave signals that fall within the model's confidence bounds. The stochastic FE model is capable of generating data for multiple damage cases and varying temperatures, enriching the training dataset for 1D CNNs. By incorporating insights from the stochastic FE model during the training phase, the machine learning algorithm can leverage the model's inherent uncertainty and improve generalization across unobserved experimental data.

1.6 OUTLINE

This thesis is structured into the following chapters:

- **Chapter 1 - INTRODUCTION**: This chapter covers the motivation, objectives, main contributions, and scope of the Thesis.
- **Chapter 2 - A BRIEF STATE-OF-THE-ART REVIEW**: This chapter presents a brief literature review concerning SHM inspections using ultrasonic waves, neural networks for Lamb wave signal processing, and model updating techniques.
- **Chapter 3 - DAMAGE LOCALIZATION USING LAMB WAVES AND 1D CONVOLUTIONAL NEURAL NETWORKS**: This chapter begins by introducing the LaWaDe system developed by the Group of Acoustics and Vibrations (GRAVi), followed by detailing both physical and software modifications implemented to automate pitch-catch measurements. It then describes the collection of Lamb wave signals using this system for two different case studies involving simulated damage on a composite plate. Subsequently, the chapter presents an application for both global and local damage localization using a novel 1D CNN approach and compares the results with a traditional method employing DIs and an MLP network. Finally, the chapter provides practical guidelines for determining the CNN structure and concludes with final remarks.

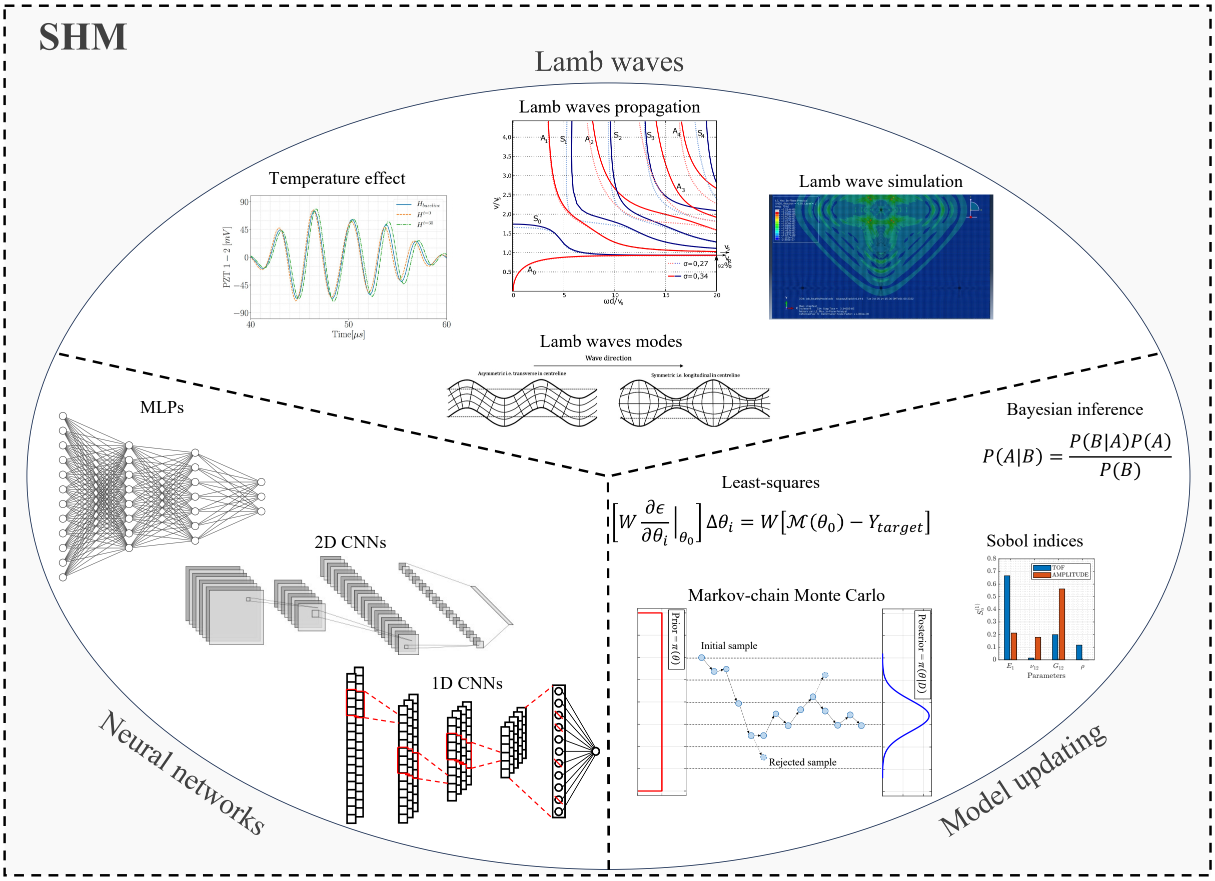
- **Chapter 4 - BAYESIAN UPDATING FOR A LAMB WAVE MODEL UNDER VARYING TEMPERATURE**: This chapter introduces a proposed framework for updating FE models for Lamb wave propagation in composite structures. It then delves into the Bayesian inference strategy, detailing the MCMC/Metropolis-Hastings algorithm used to update the probability density functions of key model parameters. This process results in a stochastic FE model that accounts for experimental uncertainty. The chapter also proposes a novel surrogate modeling strategy using artificial neural networks to decrease the computational time required for the MCMC process. Furthermore, it showcases an experimental application of the proposed framework on a carbon plate under varying temperatures, illustrating the steps and benefits of creating a stochastic numerical model. Finally, final remarks are addressed.

- **Chapter 5 - BAYESIAN DATA-DRIVEN FRAMEWORK FOR SHM UNDER LIMITED DATA**: This chapter expands the updating framework introduced previously to encompass the identification of damage parameters within an experimental setup featuring simulated damage. Employing Bayesian inference through MCMC simulations in conjunction with the Metropolis-Hastings algorithm, the framework updates an FE model by integrating observed uncertainties on experimental data. The updated model is then employed to generate supplemental data for the actual experimental data. A 1D CNN is subsequently trained on this hybrid dataset to analyze Lamb wave signals for damage assessment. Comparative analysis of different training strategies indicates that the hybrid approach provides the most accurate damage size predictions, surpassing those obtained using solely experimental data or exclusively numerically generated data. The chapter concludes with final remarks.

- **Chapter 6 - FINAL REMARKS**:This chapter comprises a discussion on the contributions to the literature, main conclusions, and path forward for future work.

2 A BRIEF STATE-OF-THE-ART REVIEW

This Chapter extends some discussions started in the motivation of the Thesis with a brief literature review on the state-of-the-art in SHM using Lamb waves, neural networks algorithms for damage diagnosis and strategies for numerical models updating. To facilitate the Chapter's flow, Fig. 4 presents a graphical abstract with the main presented themes.



Source: Prepared by the author.

Figure 4 – Graphical abstract.

- **SHM:** This Section presents the definition of SHM and exemplifies available techniques for structures inspection.

- **Lamb waves:** This Section presents the general formulation of Lamb waves, their propagation modes, and the influence of temperature on Lamb wave propagation. Additionally, it covers numerical techniques for Lamb wave simulation, and methods available for damage detection using this type of ultrasonic wave.

- **Neural networks:** This Section consists of a theoretical background of how neural networks are built, especially MLP and CNN, followed by a summary of applications of these algorithms in the field of SHM. Special attention is given to recent applications of CNNs. A list of the main works, along with their methodologies using 2D and 1D CNNs, is provided, and possible research opportunities are presented.

- **Model updating:** This Section presents methods available for updating models using experimental observations. It presents Sobol indices for sensitivity analysis, methods available for substituting a numerical model with a surrogate model, and both deterministic and stochastic updating strategies for updating model parameters.

2.1 STRUCTURAL HEALTH MONITORING

SHM has evolved from heuristic condition assessments, up to offering real-time monitoring systems that detect early anomalies or damage in aerospace, civil, or mechanical structures. According to Giurgiutiu (2014), an SHM system can be classified as either passive or active. A passive system relies on measuring operational factors to determine the state of the structure based on them. For instance, in an aircraft, one might monitor parameters such as speed, vibration levels, turbulence levels, and then employ algorithms to assess the structural state based on these metrics. In contrast, active systems utilize sensors and actuators distributed across the structure aiming to detect the presence and extent of damage. The premise of an active SHM system are similar to those of NDI systems, although there is a distinguishing feature: SHM systems are geared towards the permanent installation of sensors within the structure, allowing for analyses whenever required.

As described by Farrar, Doebling and Nix (2001), damage entails changes compromising the structure's performance and safety. Two core SHM methods exist: (i) the physics-based approach, using inverse problem techniques and finite element (FE) models to understand physical relations, and (ii) the data-based approach, which relies on machine learning algorithms to interpret structural behavior from past data, mainly when physics-based solutions are impractical due to accuracy or efficiency limits. Available algorithm options include K-nearest neighbor (SARMADI; KARAMODIN, 2020), support vector machines (SVM) (MECHBAL; URIBE; REBILLAT, 2015), and artificial neural networks (ANNs) (LEE; JANG; PARK, 2022). In the domain of SHM of plate-like structures, a particular focus has been placed on assessing structural conditions using Lamb waves.

2.2 LAMB WAVES

This Section discusses the theory behind a specific type of guided wave, known as the Lamb waves (2.2.1), and the influence of temperature on these waves (2.2.2). It also covers available techniques for simulating Lamb waves (2.2.3), and their applications in SHM (2.2.4).

2.2.1 General formulation

Guided waves are ultrasonic mechanical waves that propagate within structures and remain confined within their boundaries. Lamb waves, also known as plate guided waves, are a type of guided wave that propagate between two parallel free surfaces of a thin-walled structure. This property allows Lamb waves to travel long distances, and enables their application in ultrasonic inspections of aircraft, missiles, pressure vessels, storage tanks, and more (GIURGIUTIU, 2014). Lamb waves are multimodal in nature. At any given frequency, there are at least two modes that coexist, leading to multiple wave packets in the acquired signal. Lamb waves are also dispersive, meaning that the velocity of each mode varies with respect to its frequency. Added to the multimodal nature, this property makes the acquired signal complex in appearance. Therefore, analysis and implementation of Lamb waves are more challenging than traditional ultrasonic NDI techniques, as ultrasound inspection (MITRA; GOPALAKRISHNAN, 2016).

Generally, the analysis of wave propagation begins with the derivation of governing differential elasto-dynamic equations along with their associated boundary conditions. Frequently, these equations are transformed into the frequency-wavenumber domain to discern the dispersion characteristics. Dispersion relations, specifically the variation of group/phase velocity with frequency, delineate the nature of wave propagation. This includes the identification of propagating modes and the determination of wave velocity as a function of frequency. Beyond determining dispersion characteristics, the governing equations are also resolved in the time domain. This allows for the examination of wave amplitudes, reflections from edges and boundaries, and interactions with damages. Methods for Lamb wave simulations are discussed in Section 2.2.3.

A complete theoretical discussion of Lamb waves propagation can be found in text books on wave propagations (CHEEKE, 2017; GIURGIUTIU, 2014; LEVY, 2003), and Wang and Yuan (2007a) presents a comprehensive discussion about the formulation behind Lamb wave propagation in composite structures. However, to discuss the multimodal and dispersive natures of Lamb waves, Eqs. 1 and 2 presents the general description of these waves on a plate, which are known as the Rayleigh-Lamb equations:

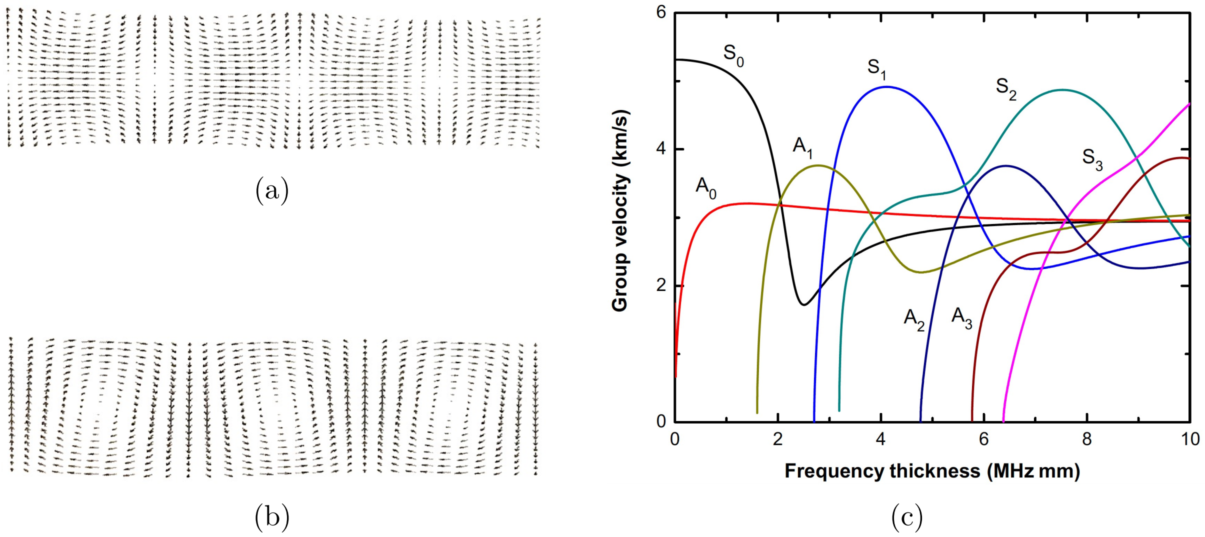
$$\frac{tan(qh)}{tan(ph)} = -\frac{4k^2qp}{(k^2-p^2)^2}, \qquad \text{(Symmetric modes)}, \qquad \text{and,} \tag{1}$$

$$\frac{tan(qh)}{tan(ph)} = -\frac{(k^2-p^2)^2}{4k^2qp}, \qquad \text{(Antisymmetric modes)} \tag{2}$$

where $p^2 = \frac{\omega^2}{c_L^2} - k^2$, $q^2 = \frac{\omega^2}{c_T^2} - k^2$, and $k = \frac{2\pi}{\lambda}$. In these expressions, h represents the half thickness of the plate; k, ω and λ are wavenumber, circular frequency, and wavelength of the wave, respectively; and c_L and c_T are the longitudinal and transverse wave velocities of the bulk material comprising the plate, respectively. Eqs. 1 and 2 describes that Lamb waves can exist in two basic types of modes: symmetric and antisymmetric. Symmetric modes are labeled as S0, S1, S2,..., whereas antisymmetric modes are referred to as A0, A1, A2,... Symmetric modes correspond to axial propagation modes, while antisymmetric modes are equivalent to bending propagation modes. Figures 5a and 5b provide a cross-sectional representation of a structure's thickness and the displacement field of each mode at a specific instant in time. Figure 5c presents the wave group velocity disperion curves for a plate made of A606 steel.

Note that at low frequency-thickness values in Fig. 5c, the S0 mode velocity exceeds that of A0, making them relatively easy to differentiate. However, as the frequency-thickness value approaches 1.6 MHz.mm, the velocity of the S0 mode decreases while that of the A0 mode increases, leading to an inversion of their velocities and the emergence of the A1 mode. Traditional techniques for damage detection using Lamb waves typically focus on the lower region of the frequency-thickness chart, as they often depend on the separation of wave modes (IHN; CHANG, 2008; HU *et al.*, 2010; NAGY; SIMONETTI; INSTANES, 2014).

Lamb waves can be generated through various methods, including the use of laser,



(a)

(b)

(c)

Source: Adapted from and Yuan (2016).

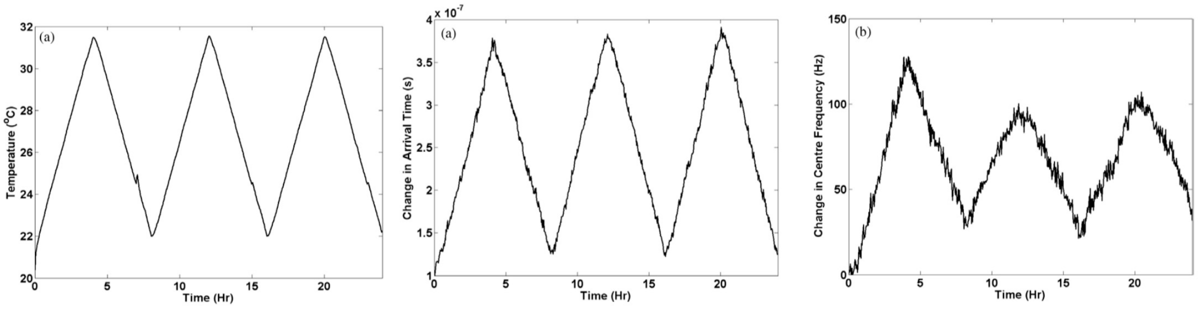
Figure 5 – Lamb wave modes: (a) displacement field of symetrical mode; (b) displacement field of anti-symetrical mode; and (c) group velocity dispersion curves in an A606 steel plate structure.

ultrasound, electromagnetic acoustic transducers, and Lead Zirconate Titanate (PZT) transducers. PZTs, in particular, are widely used due to their practical benefits; they are compact, energy-efficient, and can be permanently attached to structures for continuous monitoring (PARK *et al.*, 2006; GIURGIUTIU, 2014). These sensors deform when subjected to an electric voltage, thereby inducing an acoustic vibration in the structure. As these waves are sensitive to modifications in the propagation medium, the structure's condition can be inspected by evaluating changes in the measured waves.

2.2.2 Temperature influence on ultrasonic waves

In addition to challenges related to the inherent complexity of guided wave propagation, one of the primary factors complicating the use of SHM systems based on this type of wave is their sensitivity to changes in environmental and operational conditions (MITRA; GOPALAKRISHNAN, 2016). Factors such as humidity, temperature, and boundary conditions affect the guided wave propagation. Temperature variations are among the most influential environmental aspects affecting guided waves applications, impacting propagation characteristics such as wave speeds and amplitude ratios of different wave modes. It affects the wave response through changes in the propagation medium, such as thermal expansion/contraction, and through modifications in the transducers (MARZANI; SALAMONE, 2012).

The study conducted by Konstantinidis, Drinkwater and Wilcox (2006) addresses the



(a) (b) (c)

Source: Adapted from Konstantinidis, Drinkwater and Wilcox (2006)

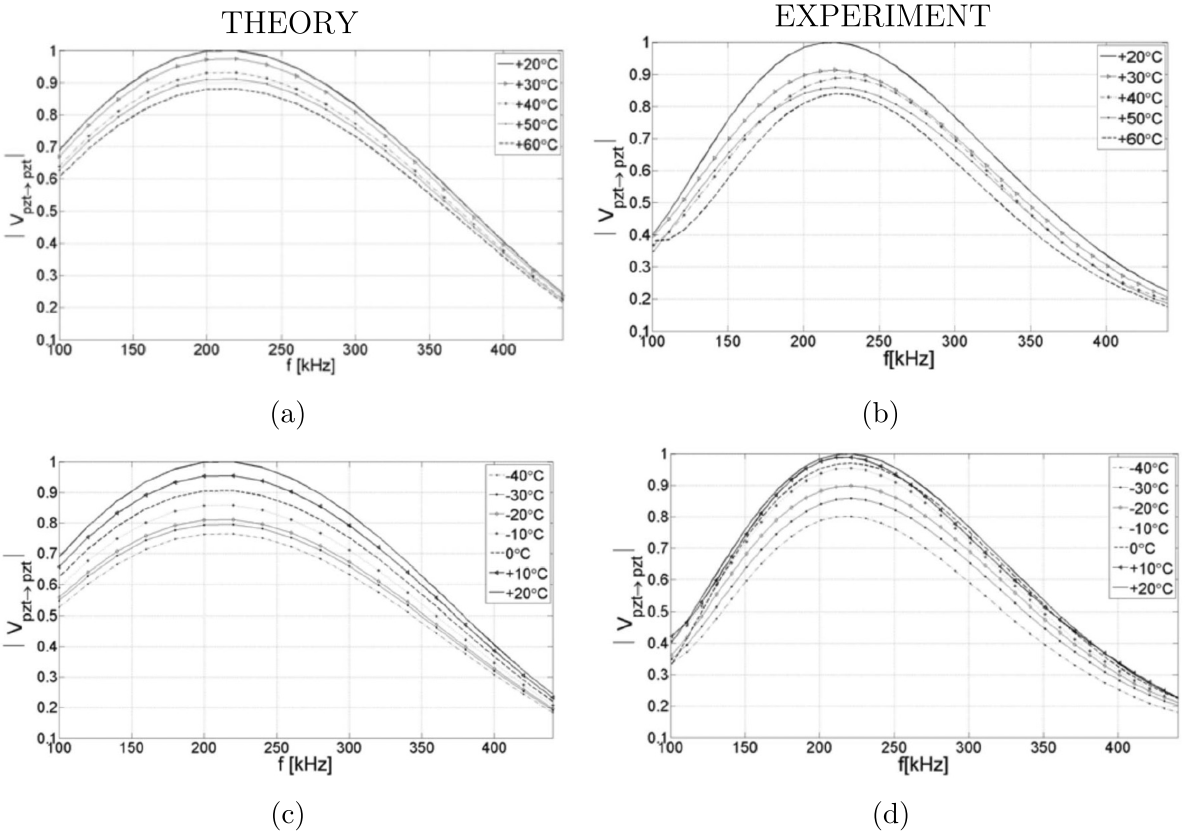
Figure 6 – Temperature influence on Lamb wave propagation: (a) temperature cycle; (b) change in wave package arrival time; and (c) change in central frequency.

influence of temperature variations on SHM evaluations with Lamb waves. The authors performed an experiment with two PZT sensors on an aluminum plate. To assess the influence, they varied the specimen's temperature between 22°C and 32°C, as depicted in Fig. 6a. The study's findings show that temperature influences propagation speed, signal amplitude, and frequency. Figure 6b illustrates the arrival time variation of the A0 wave packet, and Fig. 6c shows the variation in central signal frequency. In methods based on amplitude relations or the group speed of wave packets, these variations can lead to misinterpretations in both damaged and undamaged structures, resulting in false negatives and false positives.

In the study by Konstantinidis, Drinkwater and Wilcox (2006), the temperature range assessed spanned only 10 degrees. A broader temperature range was examined by Scalea and Salamone (2008). The authors varied the temperature between -40°C and +60°C, corresponding to the operational range of commercial aircraft. They developed an analytical model of the interaction between the PZT sensor and an aluminum plate with the temperature as variable. This model was subsequently validated with experimental results. Figure 7 shows the normalized amplitude variation of the S0 wave packet at temperatures below ambient (-40°C to +20°C) and above ambient temperature (+20°C to +60°C). The data indicate that amplitude variation can reach up to 20% at temperatures lower than the initial ambient temperature.

The Baseline Signal Stretch (BSS) and Optimal Baseline Subtraction (OBS) methods have been proposed by several authors to compensate for temperature effects, including Konstantinidis, Wilcox and Drinkwater (2007) and Croxford *et al.* (2010).

In the BSS method, the baseline signal is adjusted by either compressing or stretching it using a stretching factor to align it with the signal under analysis. This factor is



(a)

(b)

(c)

(d)

Source: Adapted from and Scalea and Salamone (2008).

Figure 7 – Pitch-catch response to S0 Lamb mode in a 1.58-mm-thick aluminum plate for the PZT actuator-sensor pair under changing temperature in the ranges: (a) and (b) above ambient +20 and +60°C; and (c) and (d) below ambient -40 to +20°C. Figures (a) and (c) represents theoretical results, and figures (b) and (d) represents experimental results.

determined using a cross-correlation-based criterion between the signals. A significant advantage of this method is that it requires only one signature collected at a specific temperature value, with subsequent signatures obtained by altering the baseline signature. However, its applicability is constrained to very narrow temperature ranges (±2°C, as per (KONSTANTINIDIS; WILCOX; DRINKWATER, 2007)) and does not allow for signal amplitude variations.

Conversely, the OBS method employs a large number of signatures for each temperature within the operating range, typically spaced at 1 °C intervals. The optimal signature is chosen based on an error criterion, such as the mean square deviation. While this method enables a more precise correction for temperature effects, it requires the collection of a large number of signatures, which can be challenging in practical scenarios According to Croxford *et al.* (2010), a combined approach of BSS and OBS methods provides a more robust solution for temperature compensation.

2.2.3 Ultrasonic waves simulation

Lamb wave simulation techniques have been extensively reviewed by Willberg *et al.* (2015). Among the available techniques, three methods stand out in the literature: the wave finite element (WFE) method, the Local Interaction Simulation Approach (LISA), and the FE method.

The WFE method is a semi-empirical method that employs the principles of the FE method to simulate wave propagation (MEAD, 1973). It involves discretizing the structure into finite elements, formulating the wave equation, and solving it in either the frequency or time domain to analyze wave behavior. This approach is advantageous for its ability to handle complex geometries and material properties, providing detailed insights into wave phenomena such as dispersion and mode conversion (MACE; MANCONI, 2008a; THIERRY; BROWN; CHRONOPOULOS, 2018; APALOWO; CHRONOPOULOS, 2019)

The LISA, as described by Packo *et al.* (2012), is based on the finite differences (FD) method. The elasto-dynamic wave equations are converted into FD equations in time and space. LISA is effective for simulating Lamb waves in complex structures due to its ability to handle heterogeneous materials and irregular geometries. It is also computationally efficient for large-scale simulations. However, LISA's accuracy is dependent on the medium's discretization and the fidelity of local interaction rules (WILLBERG *et al.*, 2015), which must be carefully defined for each case using the FD approach.

The LISA and WFE methods stand out for their relatively modest computational demands when contrasted with the FE method. Despite this advantage, a significant limitation is their lack of standardized integration in commercial softwares. Consequently, they necessitate custom implementation for each problem being investigated.

The FE method stands out among available approaches due to its relative ease of implementation and commercial codes with extensive element libraries that are readily available. As the FE method assumes a full 3D displacement field, it can accurately capture the wavefront propagation in complex structures (LUCA *et al.*, 2020). Multiple works applied the FE method for simulating Lamb waves in aluminum (MOSER; JACOBS; QU, 1999; GRESIL *et al.*, 2013) and composite structures (YANG *et al.*, 2006; MACE; MANCONI, 2008b; NG *et al.*, 2012).

For FE models, the choice of mesh size and simulation time-step is essential for achieving accuracy and stability in elastodynamic simulations. To ensure accuracy, it is advisable to employ a minimum of 6 to 10 elements per the shortest propagating wavelength (MACE; MANCONI, 2008b). To maintain stability, adherence to the Courant-Friedrichs-Lewy condition is essential (COURANT; FRIEDRICHS; LEWY, 1967). This condition stipulates that the fastest propagating wave should not traverse more than one element in a single time-step. Satisfying these two conditions usually makes FE models for Lamb wave simulations very computationally expensive.

Only a few works address the problem of simulating Lamb waves under variable temperature conditions. The author found three works in literature addressing temperature effect on Lamb waves simulation:

- Han (2007) referred to charts from MIL-HDBK-5J to derive properties for Aluminum 2024-T3 and simulate Lamb waves in temperatures ranging from 0°C to 300°C;
- Mariani *et al.* (2021) accounted for the impact of temperature in a FE model for Lamb waves by linearly shifting the excitation signal in the time domain; and
- Perfetto *et al.* (2022) studied both experimentally and numerically the propagation characteristics of Lamb waves in a CFRP plate. The authors imposed a linear variation for all material parameters to account for varying temperature.

Among the identified studies, the effect of temperature is incorporated into numerical simulations by either linearly modifying the mechanical properties of materials or the excitation signal. Therefore, the implementation of reliable FE models for simulating Lamb waves that include experimental uncertainties remains an open research field.

2.2.4 Damage detection using Lamb waves

Damage detection and localization techniques using Lamb waves are broadly categorized into two groups: theoretical model-based methods and data-driven approaches. The former involves assessing structural degradation by integrating theoretical analysis with experimental data, and the latter essentially process and interpret the captured wave signals. Theoretical techniques often evaluate metrics like degradation, stiffness reduction, and energy dissipation. However, a significant drawback of these methods is the need for specific adjustments and reanalysis for each application, limiting their generalization capabilities.

Data-driven techniques apply signal processing and identification techniques to extract information from Lamb wave signals. By assessing the behavior of Lamb waves in an undamaged structure, one can obtain a "signature" (often referred to as a Baseline). Subsequent measurements are compared with this baseline, and damage detection techniques can be employed to locate and quantify damage. Among data-driven approaches, there are several algorithms to interpret Lamb wave information, including support vector machines (SVM) (MECHBAL; URIBE; REBILLAT, 2015), autoregressive (AR) models (SILVA, 2018), Gaussian process (PAIXAO *et al.*, 2021), K-nearest neighbor (SARMADI; KARAMODIN, 2020), and neural networks (SU; YE, 2004; De Fenza; SORRENTINO; VITIELLO, 2015; AZUARA; RUIZ; BARRERA, 2021; ZHANG *et al.*, 2022; LOMAZZI *et al.*, 2023). These algorithms are often supported by signal pre-processing techniques, including time of flight (TOF) estimation (STASZEWSKI, 2002), Hilbert transform (SU *et al.*, 2019a), Fourier transform (HORA; ČERVENÁ, 2012), and wavelet transform analysis (FENG; RIBEIRO; RAMOS, 2018). Section 2.3.3 addresses neural networks algorithms for Lamb waves processing.

2.3 NEURAL NETWORKS

Artificial intelligence emerged as a branch of computer science in the 1950s. Since then, it has produced tools with potential applications in engineering to solve problems that require human intelligence (Pham and Pham, 1999). Among the available algorithms, artificial neural networks (ANNs) have gained prominence due to their ability to generalize, handle non-linearities, and extrapolate knowledge from an incomplete dataset. A neural network is a computational model loosely inspired on the brain of mamals. Models based on neural networks are composed of basic units called neurons, which are interconnected and perform parallel calculations (GOODFELLOW; BENGIO; COURVILLE, 2016).

In terms of structure, neural networks can be divided into two types: feedforward and recursive. In feedforward networks, all computation flows in one direction, from the input to the output, with the output at any given moment depending solely on the input at that moment. In contrast, recursive networks feature outputs from some neurons feeding back into earlier layers, endowing these networks with dynamic memory; thus, the output at any given time depends on both the current input and previous information.

The learning process of a neural network is called training, which can be supervised or unsupervised. In supervised training, the neural network is exposed to a series of input data and the expected output. Based on the provided data, the network adapts and adjusts its parameters, storing the knowledge in the weights of the connections between the neurons. In unsupervised training, the network only receives input data and learns by grouping data into classes with common characteristics. After training, the network's efficiency is evaluated with a new, previously unseen dataset, with its performance measured by its accuracy with this data.

Throughout this work, two architectures of neural networks are utilized, the multilayer perceptron and the convolutional neural network. In the following subsections, the theoretical bases of both architectures are examined (Sections 2.3.1 and 2.3.2), followed by a comprehensive summary of their respective applications in the field of SHM (2.3.3).

2.3.1 Multilayer perceptron

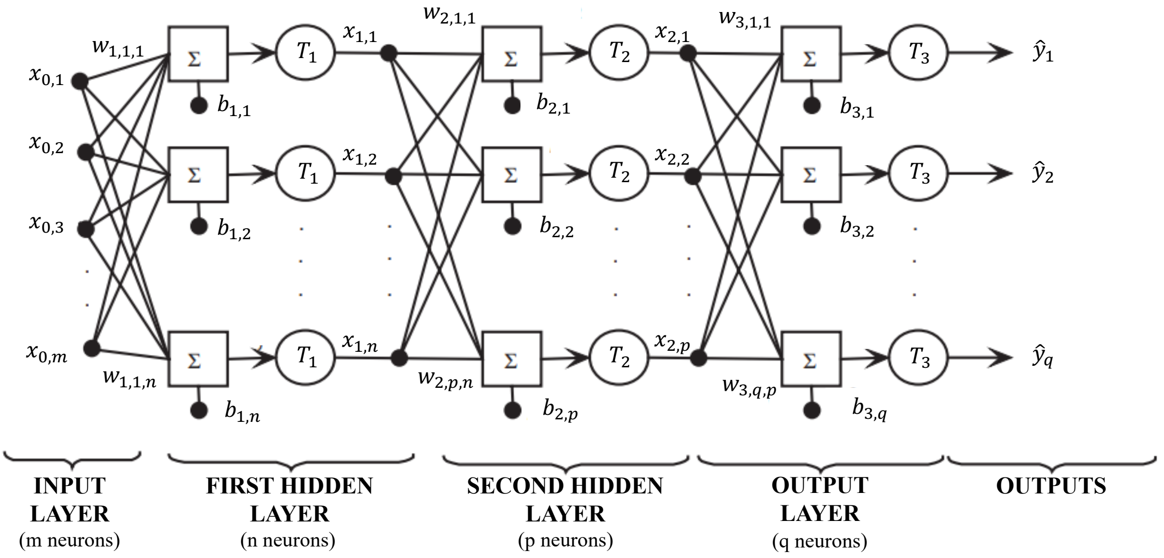
According to Pham and Pham (1999), the most commonly used neural network type is the multilayer perceptron (MLP) due to its ease of implementation and robustness. MLP networks consist of a series of interconnected layers of neurons that receive the information from the previous layer neurons, process it, then pass it forward.

Assuming a training dataset with dimension $M \times N$, where M is the number of points per array, and N is the number of samples in the dataset, the l-th sample vector is represented by $\boldsymbol{x}^{(l)} \in \mathbb{R}^{M \times 1}$. In a supervised learning scheme, the expected output for a training sample is represented by $\boldsymbol{y}^{(l)} \in \mathbb{R}^{P \times 1}$, where P is the number of outputs. The objective of the neural network is to map the function $\mathcal{F} : \boldsymbol{x}^{(l)} \rightarrow \boldsymbol{y}^{(l)}$. A neural network with at least one hidden layer can act as a universal approximator (HORNIK; STINCHCOMBE; WHITE, 1989). Each neuron in a hidden layer performs part of this mapping in the form:

$$\boldsymbol{x}_{i+1,k} = \boldsymbol{\mathcal{T}}\left(\sum_{j=1}^{m} w_{ij,k} x_{ij} + b_{i,k}\right). \tag{3}$$

Considering the kth unit (or neuron) of the ith layer, it receives the output x_{ij} from each jth unit of the $(i-1)$th layer. The values x_{ij} are then multiplied by a weight w_{ijk}, and these products are added. A bias $b_{i,k}$ is added to the result, and then an activation function ($\boldsymbol{\mathcal{T}}$) is applied to the result that is propagated to the next layer. The activation function is responsible for adding non-linearity to the system, and it can assume multiple forms, from Rectified Linear Unit (ReLU) and leaky ReLU functions, to sigmoid functions, e.g., logistic function and *tanh*.

The mapping of the inputs (or features) $\boldsymbol{X}$ to the outputs (or labels) $\boldsymbol{Y}$ is achieved



Source: ANN structure adapted from De Fenza, Sorrentino and Vitiello (2015).

Figure 8 – Structure of a feedforward MLP network.

through the training process, in which the parameters $\boldsymbol{\Theta}_{NN}$ of the network $\boldsymbol{\mathcal{N}}$ are updated using a set of sample pairs $(\boldsymbol{X}^{(l)}, \boldsymbol{Y}^{(l)})$ through the minimization of a loss function $\boldsymbol{\mathcal{L}}(\boldsymbol{\Theta}_{NN}) = \boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{N}}(\boldsymbol{X}, \boldsymbol{\Theta}_{NN}) - \boldsymbol{Y})$. The choice of loss function depends on the task at hand: cross-entropy is typically used for classification tasks, while the mean squared error (MSE) is favored for regression tasks.

Algorithm 1 outlines the standard procedure for training a neural network. Available data is usually divided in three datasets: training, validation and test. The training dataset is used to adjust the weights of the network, essentially "teaching" it the desired behavior. The validation dataset, separated from the training data, is utilized to evaluate the model during training, allowing for tuning of hyperparameters. Finally, the test dataset, not used during training, is employed to assess the performance of the fully trained model, offering an unbiased evaluation of its effectiveness. Initially, network weights are assigned small random values to break symmetry and ensure that gradients are neither too small nor too large at the onset of training, as suggested by He *et al.* (2015b). During forward propagation, the network generates predicted outputs, which are then compared to the actual outputs to compute the loss. Backpropagation (RUMELHART; HINTON; WILLIAMS, 1986) follows, where network parameters are updated. This involves computing the gradients of the loss function with respect to the network parameters and applying an optimization algorithm to adjust the weights in the direction of the negative gradient, commonly referred to as gradient descent. Popular optimization algorithms include Stochastic Gradient Descent (SGD), SGD with momentum (SGDM), and Adaptive Moment Estimation (Adam). The magnitude of weight updates during each training pass is controlled by a hyperparameter known as the learning rate, denoted by α. The learning rate is a critical element that determines the step size at each iteration while moving toward a minimum of the loss function. A higher learning rate can indeed accelerate the training process by permitting larger updates to the weights. However, if the learning rate is set excessively high, it may render the training process unstable, potentially causing the optimization algorithm to overshoot minima or even diverge, failing to converge to an optimal solution. Conversely, an overly low learning rate may lead to a slow convergence process. The learning rate is typically selected based on empirical evidence and experimentation and is often adjusted dynamically in response to the neural network's performance on validation data. The training process progresses through multiple epochs, continuing until it meets the predefined stopping criteria.

Algorithm 1: Training steps of a neural network with gradient descent

```
Result: Trained Neural Network
Input: Training data ($\boldsymbol{X}^{(Tr)}, \boldsymbol{Y}^{(Tr)}$), validation data ($\boldsymbol{X}^{(Val)}, \boldsymbol{Y}^{(Val)}$), network architecture
       $\boldsymbol{\mathcal{N}}$
Initialize network weights $\boldsymbol{\Theta}_{NN}$;
while Stopping criterion not met do
    Shuffle the training data ($\boldsymbol{X}^{(Tr)}, \boldsymbol{Y}^{(Tr)}$);
    Divide the training data into mini-batches;
    foreach mini-batch do
        Perform a forward pass through the network;
        Calculate the loss (e.g., cross-entropy loss or );
        Perform a backward pass to compute gradients (backpropagation);
        Update weights with gradients (e.g., using SGD, SGDM, Adam);
    end
    Evaluate performance on validation set;
    If performance satisfies the criterion, or no improvement, stop training;
end
```

Source: Prepared by the author.

To manage computational resources more efficiently and to improve convergence, the training data is typically divided into smaller subsets known as mini-batches. This approach, known as mini-batch training, allows for more frequent updates of the network parameters, contributing to a smoother and often faster convergence process. After training, the performance of the network is evaluated on a separate test set to gauge its generalization capabilities.

As the training process consists of a multivariate optimization, $\boldsymbol{\Theta}_{NN}$ is not necessarily unique and can produce multiple networks that fit the data. To encode the preference for certain sets of weights and remove part of this ambiguity, it is common to add a regularization loss in the form of: $\mathcal{R}(\Theta_{NN}) = \lambda \sum_i \sum_j (\Theta_{NN})^2_{ij}$, in which $\lambda > 0$ is a regularization parameter. This is commonly called L_2 regularization, and it induces the optimization algorithm to select models with smaller parameters. This regularization strategy also impacts reducing overfitting, as highly specialized networks with concentrated parameters are penalized over the ones with a more uniform weight distribution.

2.3.2 Convolutional neural networks

CNNs are feedforward neural networks inspired by animals' visual cortex. The first version of a CNN was proposed by Fukushima and Miyake (1982), based on the receptive fields of monkeys' visual cortex. The fundamental difference between a convolutional network and a traditional network is that the former employs the convolution operation instead of multiplication in at least one of its layers. CNNs have become dominant in the deep learning field when it comes to tasks like object recognition in images and videos and voice recognition (KRIZHEVSKY; SUTSKEVER; HINTON, 2012a).

Acording to Kiranyaz *et al.* (2021), the popularity of CNNs can be attributed to the following advantages:

- CNNs fuse the feature extraction and feature classification processes into a single learning body. They can learn to optimize the features during the training phase directly from the raw input;
- Since CNN neurons are sparsely-connected with tied weights, CNNs can process large inputs with a great computational efficiency compared to the conventional fully-connected MLP networks;
- CNNs are robust to small transformations in the input data including translation, scaling, skewing and distortion;
- CNNs can adapt to different input sizes.

According to Abdeljaber *et al.* (2017), a typical CNN is a feedforward network that comprises a series of convolutional and subsampling layers. The network operates by conducting convolution operations between a kernel and the input data in each layer, followed by subsampling to reduce the spatial dimensions before forwarding it to the next layer. They are capable of detecting patterns in input data such as images or time series with minimal preprocessing required prior to input, and they are classified according to the input layer. In 2D networks, two-dimensional tensors, such as images, are used as input, while in 1D networks, 1D tensors, such as time series, are used. In both cases, the input signals may have multiple parallel channels, such as RGB channels in images or multiple parallel series in the case of 1D time series.

A CNN performs a similar mapping $\mathcal{F}: \boldsymbol{x}^{(l)} \to \boldsymbol{y}^{(l)}$ as described for the MLP network, but it uses a sequence of convolution and pooling operations. Throughout this work, network layers are named using the convention I (Input), FC (Fully-Connected), C (Convolution), R (ReLU), P (Pooling), D (Dropout), and O (Output), respectively. And for enhanced visibility, each basic building block, i.e., convolution, pooling, and ReLU, or fully-connected and ReLu, is grouped and presented within parentheses in the naming convention. For instance, a:

$$I - (C - R - P) - D - (C - R - P) - D - (FC - R) - O \tag{4}$$

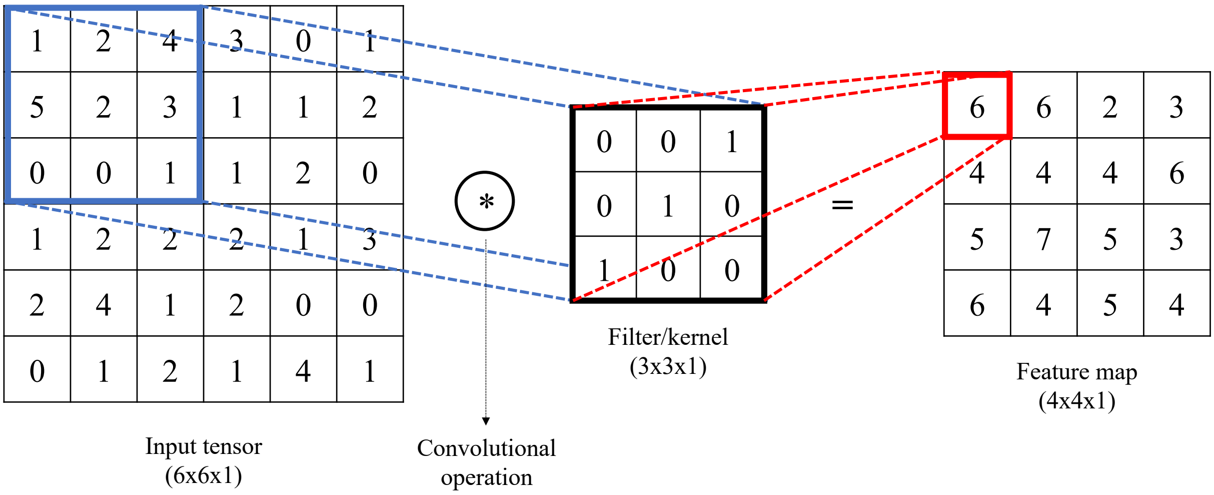
network consists of an input layer, followed by a convolution layer with ReLU activation and pooling, a dropout layer, one more convolution-ReLU-pooling block, another dropout layer, a fully-connected layer, a ReLu layer, and finally the output layer. This structure enables the extraction of increasingly complex features from the input data. The following subsections present each of these layers.

Convolution layer

The main layer in a CNN is the convolutional layer, in which a filter or kernel is convolved with the input tensor. This operation can be represented mathematically as $\boldsymbol{Y} = \boldsymbol{X} * \boldsymbol{W}$, where $\boldsymbol{Y}$ is the output feature map obtained by convolving the kernel $\boldsymbol{W}$ with the input tensor $\boldsymbol{X}$. Figure 9 depicts the convolution operation for a $6 \times 6 \times 1$ input tensor with a $3 \times 3 \times 1$ kernel. The resulting feature map is a tensor with dimensions $4 \times 4 \times 1$.

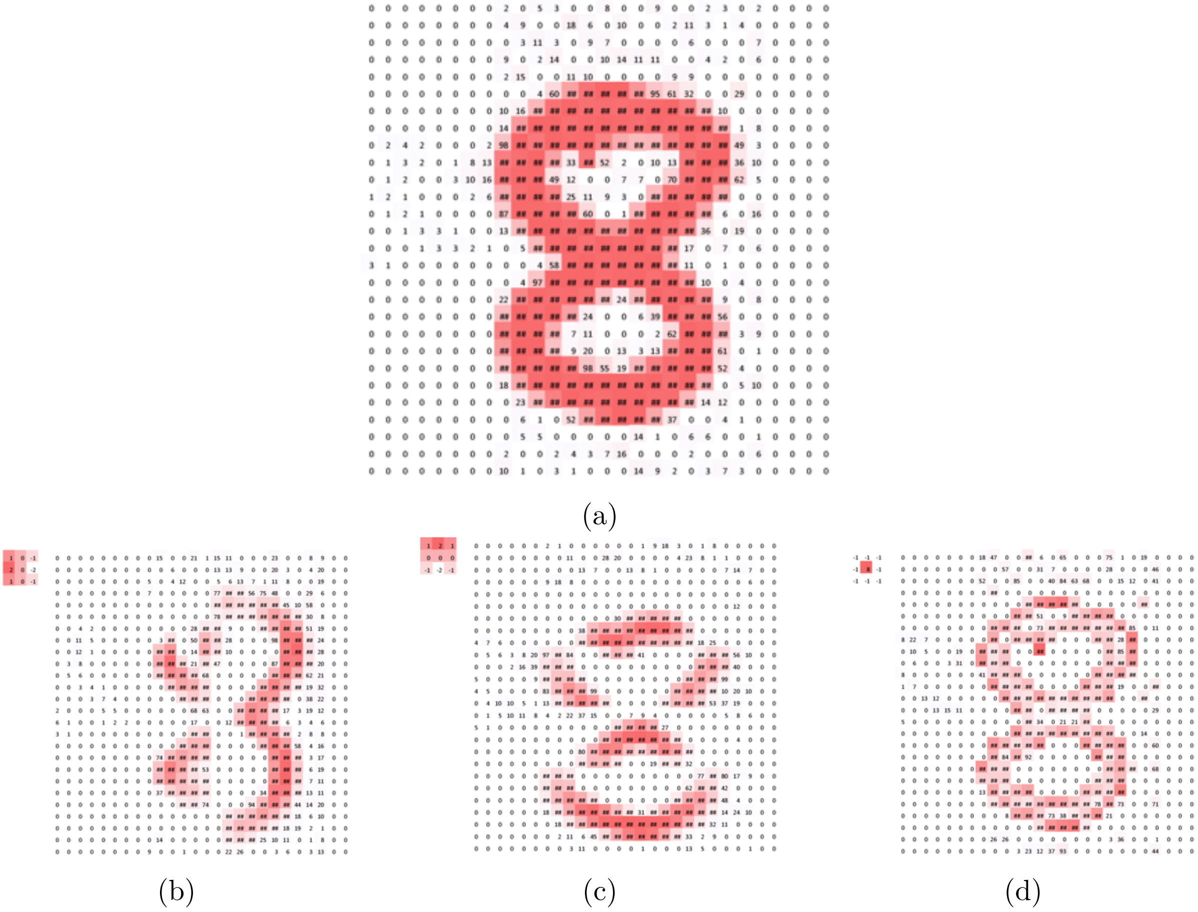
It is important to notice that the kernel depth matches the last dimension of the input tensor (one in Fig. 9), while the resulting feature map has always one as last dimension. Howerver, multiple kernels can be used as different feature extractors, as depicted in Fig. 10. The resultant feature maps for all kernels are stacked and passed to subsequent layers.

Besides kernel size, the convolutional layer has other two parameters: stride and padding. Padding refers to the addition of extra points (or pixels in the case of images) around the border of the input volume or feature map. The primary purpose of padding is to control the spatial size of the output feature maps. Commonly, "zero padding" is



Source: Prepared by the author.

Figure 9 – Demonstration of convolutional operation.


(a)

(b)

(c)

(d)

Source: Adapted from Yamashita *et al.* (2018).

Figure 10 – Examples of how kernels in convolutional layers extract features from an input tensor: (a) input tensor; and multiple kernels working as different feature extractors, such as (b) vertical edge detector; (c) horizontal edge detector; and (d) outline detector (bottom). Note that red color intesity represents magnitude, each small image is kernel, and those in the right are output feature maps.

used so the convolutional filters can process the border regions effectively. Alternatively, ‘‘same padding’’ maintains the size of the output feature map to be the same as the input, and ‘‘valid padding’’ refers to the absence of padding. Stride in CNNs is the step size the filter takes when moving across the input. A stride of one moves the filter pixel by pixel, yielding detailed outputs, while larger strides reduce output size and computational load by skipping pixels. After the convolution operation, a bias $\boldsymbol{b}$ is added to the result, and the result is allocated in the proper position of the feature map.

Algorithm 2 presents the convolution operation in an input array X. The dimensions of the output feature map are calculated based on the input dimensions, filter size, number of filters, stride, and padding. The algorithm then iterates over each filter and applies it to the corresponding region in the input volume. The sum of the element-wise product of the filter and the input region is calculated and assigned to the feature map region. This process is repeated for each position in the output volume and for each filter.

Algorithm 2: Forward pass through a convolutional layer in a CNN

```
Result: Feature maps after convolution
Data: Input volume $X$, convolutional filters $W$, bias $b$, stride $s$, padding $p$
Calculate output dimensions based on $X$, $W$, $s$, and $p$;
Initialize output feature maps to zero;
for each filter $w$ in $W$ do
    for each position $x, y$ in the output feature map do
        Define a region $R$ in the input image corresponding to the current $x, y$ position,
          considering stride and padding;
        Perform element-wise multiplication of $R$ and filter $w$;
        Sum the result and add bias $b$ to get a single value;
        Assign the result to the appropriate position in the output feature map;
    end
end
```

Source: Prepared by the author.

The output of a convolutional layer is passed to an activation layer, usually a ReLU function in deep learning applications. This activation function is applied element-wise to the feature map , introducing non-linearity, and resulting in an activated feature map Z. Therefore, a convolutional layer with activation function can be described as:

$$\boldsymbol{Z} = ReLU(\boldsymbol{X} * \boldsymbol{W} + \boldsymbol{b}) \tag{5}$$

in which ReLU represents the activation function.

ReLU layer

The Rectified Linear Unit (ReLU) has emerged as one of the most widely used activation functions in deep learning, despite the existence of several other activation functions such as sigmoid, tanh, and Leaky ReLU. Its popularity in deep learning networks can be attributed to its simplicity. The ReLU function operates on a simple principle: it outputs the input directly if it is positive; otherwise, it outputs zero. Mathematically, it is defined as $f(x) = max(0, x)$. This means that for any positive input, the output is identical to the input, and for any negative input, the output is zero. This activation function introduces non-linearity into the network, enabling the learning of complex patterns and relationships within the data (KRIZHEVSKY; SUTSKEVER; HINTON, 2012b).

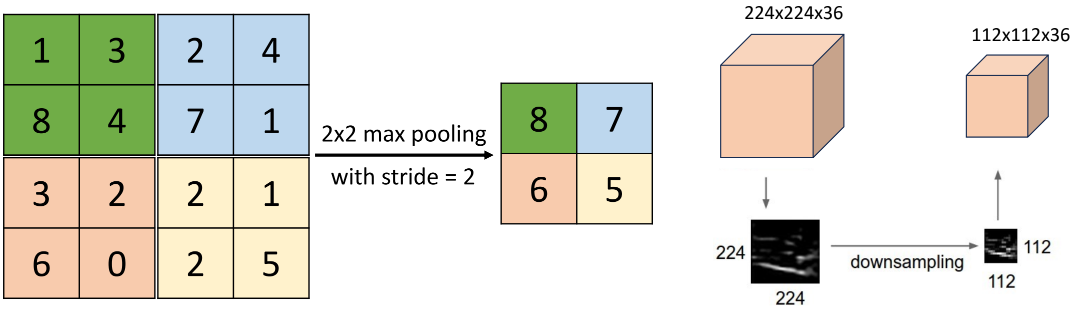
One of the primary advantages of ReLU in deep learning networks is its ability to alleviate the vanishing gradient problem. This issue occurs when gradients become too small to drive effective learning, particularly in deep networks with many layers. Since the gradient of the ReLU function is either zero (for negative inputs) or one (for positive inputs), it ensures that during backpropagation, the gradients do not diminish exponentially as they pass through multiple layers. Additionally, the ReLU function is computationally inexpensive compared to other activation functions like sigmoid or tanh, as it involves a simple max operation (SZANDALA, 2020).

Pooling layer

The pooling operation is responsible for reducing the spatial dimensions of the feature map while preserving essential information in CNNs. This operation is relevant in progressively decreasing the spatial size of the representation, which in turn reduces the number of parameters and computational load in the network.

Max pooling and average pooling are the two commonly employed pooling operations. Max pooling selects the maximum value within a pooling window, while average pooling computes the average of the values within the window. The pooling layer operates independently on each depth slice of the input and resizes it spatially. It typically employs the MAX or AVERAGE operations.

In 2D CNNs, the most common pooling layer comprises a 2x2 filters applied with a stride of 2, which downsamples every depth slice in the input by 2 along both the width and height dimensions. In this context, each pooling operation involves evaluating four



1	3	2	4
8	4	7	1
3	2	2	1
6	0	2	5

8	7
6	5

(a) (b)

Source: Prepared by the author.

Figure 11 – Demonstration of pooling layer principle: (a) Max pooling operation; and (b) dimensionality reduction.

numbers, corresponding to a small 2x2 region in some depth slice, as illustrated in Fig. 11a. This process discards 75% of the activations but retains the spatial information of the original array (see Fig. 11b). Importantly, the depth dimension of the feature map remains unchanged. For shallow networks, larger pooling filters sizes and stride can be used to speed the data subsampling from large inputs, at the cost of loss of information. Additionally, pooling layers typically do not employ zero-padding.

Normalization and dropout layers

Along the network, normalization layers, such as batch normalization, are utilized to normalize the activations within a layer to address the problem of internal covariate shift. Batch normalization applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. By normalizing the activations, these layers help stabilize and regularize the learning process, enable faster convergence, and improve the network's generalization capabilities (IOFFE; SZEGEDY, 2015).

A dropout layer can also be applied to randomly set a subset of activations to zero during training. The primary purpose of dropout is to prevent overfitting and enhance the generalization performance of the network. By dropping out neurons, dropout layers promote the development of a more robust network that does not depend too heavily on specific neuron connections (SRIVASTAVA *et al.*, 2014).

Fully connected layer

At the end of the network, the pooled feature map is flattened into a vector and fed into one or more fully connected layers, similar to an MLP network, and the output layer of the CNN applies an activation function suitable for regression or activation.

Output layer

An output layer is capable of performing either classification or regression tasks. Each task requires a distinct loss function, which assesses the discrepancy between the network's predictions and the true labels during the training phase.

- *Classification layer*

Classification layers implement the cross-entropy loss for classification tasks involving mutually exclusive classes. In standard classification networks, a softmax layer typically precedes the classification layer (BISHOP, 2006).

The softmax function of a single ith unit of a layer with $K \geqslant 1$ units is given by:

$$\hat{y}_i(\boldsymbol{z}) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{6}$$

in which $\boldsymbol{z}$ is the input of the softmax unit, $i = 1, 2, ...K$, $\boldsymbol{z} = (z_1, z_2, ...z_K) \in \mathbb{R}^K$, $0 < \hat{y}_i < 1$ and $\sum_{i=1}^{k} \hat{y}_i = 1$. The softmax function normalizes the input values of each unit between 0 and 1, and assures that the sum off all units output is equals to one.

During training, the softmax function outputs are used to assign each input to one of the K mutually exclusive classes, employing the cross-entropy function in a 1-of-K coding scheme:

$$\text{loss} = -\sum_{i=1}^{K} y_i \ln \hat{y}_i \tag{7}$$

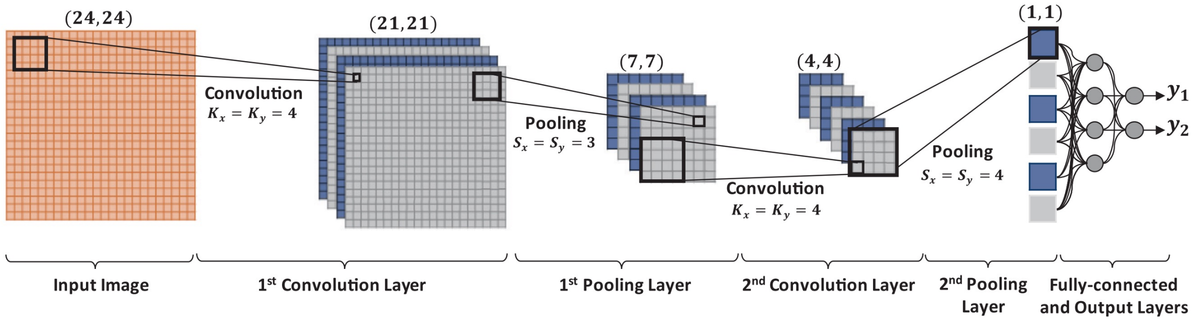
where y_i is the true label for the i-th neuron, i.e., zero or one.

- *Regression layer*

A regression layer calculates the half-mean-squared-error (MSE) loss as metric used to optimize the network. The MSE for a single observation is represented as:

$$\text{MSE} = \frac{1}{R}\sum_{i=1}^{R}(y_i - \hat{y}_i)^2 \tag{8}$$

where R is the number of responses, y_i is the target output, and $\hat{y}_i$ is the network's prediction for response i.



Source: Kiranyaz *et al.* (2021).

Figure 12 – CNN network structure with two blocks convolution-pooling.

For a network with multiple neurons in the output layer, the MSE is defined as as:

$$\text{MSE} = \frac{1}{M \times R} \sum_{i=1}^{M} \sum_{i=1}^{R} (y_i - \hat{y}_i)^2 \tag{9}$$

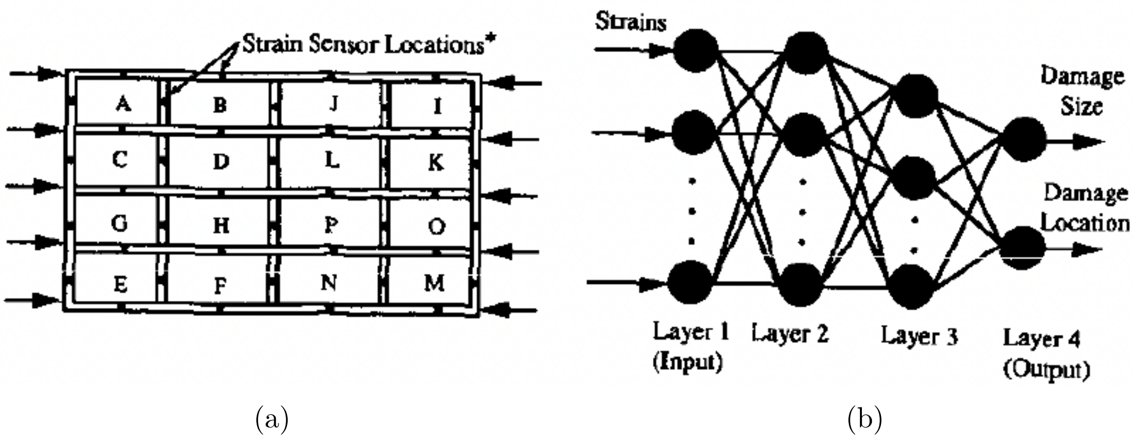
where M is the number of neurons. When training, the optimzation algorithm calculates the mean loss over the observations in the mini-batch.

Figure 12 illustrates the classical structure of a CNN. The output layer can be adapted to represent a classification or a regression layer.

2.3.3 Neural networks applications on SHM and Lamb waves

One of the pioneering works in the application of neural networks to SHM is that of Kudva, Munir and Tan (1992), conducted by researchers from the Northrop Aircraft Division, now part of Northrop Grumman Corporation. This study introduced a novel approach for detecting and quantifying significant damage in aeronautical structures, utilizing strain measurements interpreted through a neural network. Although Lamb waves were not the focus for damage detection in this instance, with strain measurements being the primary data source, the research is noteworthy for its early adoption of neural networks and utilization of numerical models in SHM.

The initial network training was carried out using strain information obtained from a finite element model of the structure. Once trained, the network was exposed to experimental strain data measured on a real structure and was expected to provide the location and size of the damage. The authors monitored strain measurements from an aluminum panel with stiffeners, as depicted in Fig. 13b, using a MLP network with two hidden layers and 40 neurons each, as shown in Fig. 13b. The panel was divided into 16 regions, named from A to P, and 40 strain gages were positioned in the center of each edge of the regions. As an applied force, a constant compression stress was applied to



Source: Adapted from Kudva, Munir and Tan (1992).

Figure 13 – One of the first applications of ANNs on the field of SHM, with (a) strain sensor locations and bays A to P; and (b) MLP that uses strain measurements to define damage size and location

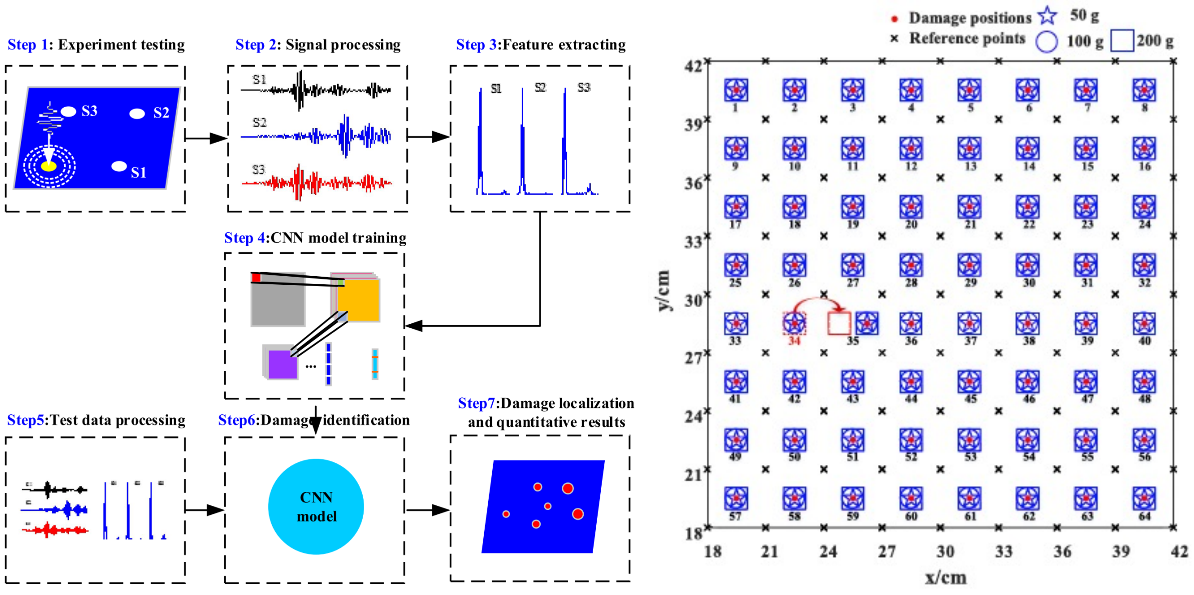
the ends of the panel. The network was trained to detect holes inside the 16 regions and inform the diameter.

The results were mixed. When the network was trained using data from a single bay, it demonstrated the capability to predict the radius. However, as the complexity of the task increased with the addition of more bays and a greater variety of possible diameters, the neural network encountered difficulties. This challenge was largely attributed to the lack of sufficient training examples. This research highlights a key drawback of neural networks: their require a lot of data and computational power for training. In practical terms, a neural network necessitates examples from the entire operational domain to effectively learn and generalize. This can be a big hurdle, especially in SHM. Introducing every conceivable type of damage into a structure for training purposes is impractical. Kudva, Munir and Tan (1992) also evaluated that employing numerical simulations for generating data is a simpler and more cost-effective alternative. Nonetheless, at the time of this study, simulating complex structures or conducting explicit simulations, as those needed for Lamb wave analysis, was not feasible.

The application of neural networks to Lamb wave detection algorithms remained underexplored in subsequent years. The computational cost and the extensive data required for training machine learning algorithms posed barriers to their widespread adoption. Moreover, it is impractical to feed an entire Lamb wave signal into a traditional MLP network due to the exponential increase in parameters within a fully connected network. Consequently, classical machine learning approaches typically necessitate preprocessing steps and depended on hand-crafted features, such as damage indices (DIs), as inputs, which could result in suboptimal classifiers and high computational demands. This focus led to the majority of research efforts being directed toward detection algorithms based on the physical principles of Lamb waves, as discussed in Section 2.2. Until the beginning of the last decade, only a handful of studies dealt with neural networks, specifically using MLPs combined with various pre-processing techniques, such as discrete wavelet transform ((SU; YE, 2004; LU *et al.*, 2009), (LIU *et al.*, 2013)), and DI calculations ((DWORAKOWSKI *et al.*, 2015), (De Fenza; SORRENTINO; VITIELLO, 2015)).

This scenario changed in the mid-2010s, as deep learning began to rapidly advance across various fields, from image processing (KRIZHEVSKY; SUTSKEVER; HINTON, 2012a), to medical applications (LITJENS *et al.*, 2017; YAMASHITA *et al.*, 2018) and natural language processing (ABDEL-HAMID *et al.*, 2012; BUBECK *et al.*, 2023). This shift can be attributed to the exponential increase in computational power, coupled with the advent of the ‘‘Big Data’’ era, where collecting and storing large volumes of data has become feasible and cost-effective. Specifically for Lamb wave applications, CNN algorithms have seen a notable surge in popularity. CNNs have a significant advantage over traditional ANNs because they can integrate feature extraction and classification tasks within a single framework (KIRANYAZ *et al.*, 2021). Unlike conventional machine learning approaches that often rely on hand-crafted features, which can be sub-optimal and computationally demanding, CNN-based methods directly extract features from the input data, thereby potentially maximizing accuracy.

The first work addressing damage detection using Lamb waves and CNNs was proposed by Su *et al.* (2019b). In this study, the authors proposed a methodology summarized in Fig. 14a . They used a square laminate (600 mm x 600 mm x 2 mm) made of carbon fiber and epoxy resin with 16 layers. The PZT mesh used had four sensors, with only one actuator and three receivers. Damages were simulated as concentrated masses positioned at 64 points on the structure. For damage severity analysis, the authors used three masses glued to the plate: 50 g, 100 g, and 200 g. The time responses were transformed into the frequency domain, resulting in a vector of 300 points for each sensor, totaling 900 points. This vector was then rearranged into a 30 x 30 matrix and provided as input to a 2D convolutional neural network. For training, a total of 140 signals were collected for each damage at each position, totaling $(64 \times 3 \times 140)$ samples. The neural network consisted of a CNN with 7 layers, with the configuration $(I - C1 - P1 - C2 - P2 - F - O)$. The network was trained with all collected data. The model validation was carried out using damages in the same training positions, just with a different collection. Figure 14b shows the results provided by the network. According


Step 1: Experiment testing
S3
S2
S1
Step 2: Signal processing
S1
S2
S3
Step 3:Feature extracting
S1
S2
S3
Step 4:CNN model training
...
Step5:Test data processing
Step6:Damage identification
CNN model
Step7:Damage localization and quantitative results
Damage positions
Reference points
50 g
100 g
200 g
y/cm
x/cm
42
39
36
33
30
27
24
21
18
18
21
24
27
30
33
36
39
42

(a) (b)

Source: Adapted from Su *et al.* (2019b).

Figure 14 – 2D CNN damage detection method: (a) signals pre-processing and training methodology; and (b) damage localization used at training (blue) and CNN predictions (red).

to the authors, out of 192 tests, only the 200 g damage at position 35 was not correctly classified, totaling 191 correct answers, or a success rate of 99.5%. However, it should be noted that the authors performed the training and testing of the network using practically the same data, since, all else being constant, the only difference between the training and validation acquisitions is the noise, which was filtered using Wavelet transform. Therefore, the high success rate can be interpreted as success with data similar to training.

After the work of Su *et al.* (2019b), there have been multiple articles addressing Lamb waves and CNNs in multiple SHM areas, such as fatigue crack detection ((LIU; ZHANG, 2019b), (XU *et al.*, 2019)) , corrosion in metallic plates (ECKELS *et al.*, 2022), bridge decks inspection (SHI *et al.*, 2023), weak bonds inspection (RAMALHO *et al.*, 2023), pipelines inspections (SHANG *et al.*, 2023), impact detection (YANG *et al.*, 2023), among others. These works used CNNs either as feature extractors or as image classifiers, applying a transformation to the time series to generate an input image for a 2D CNN.

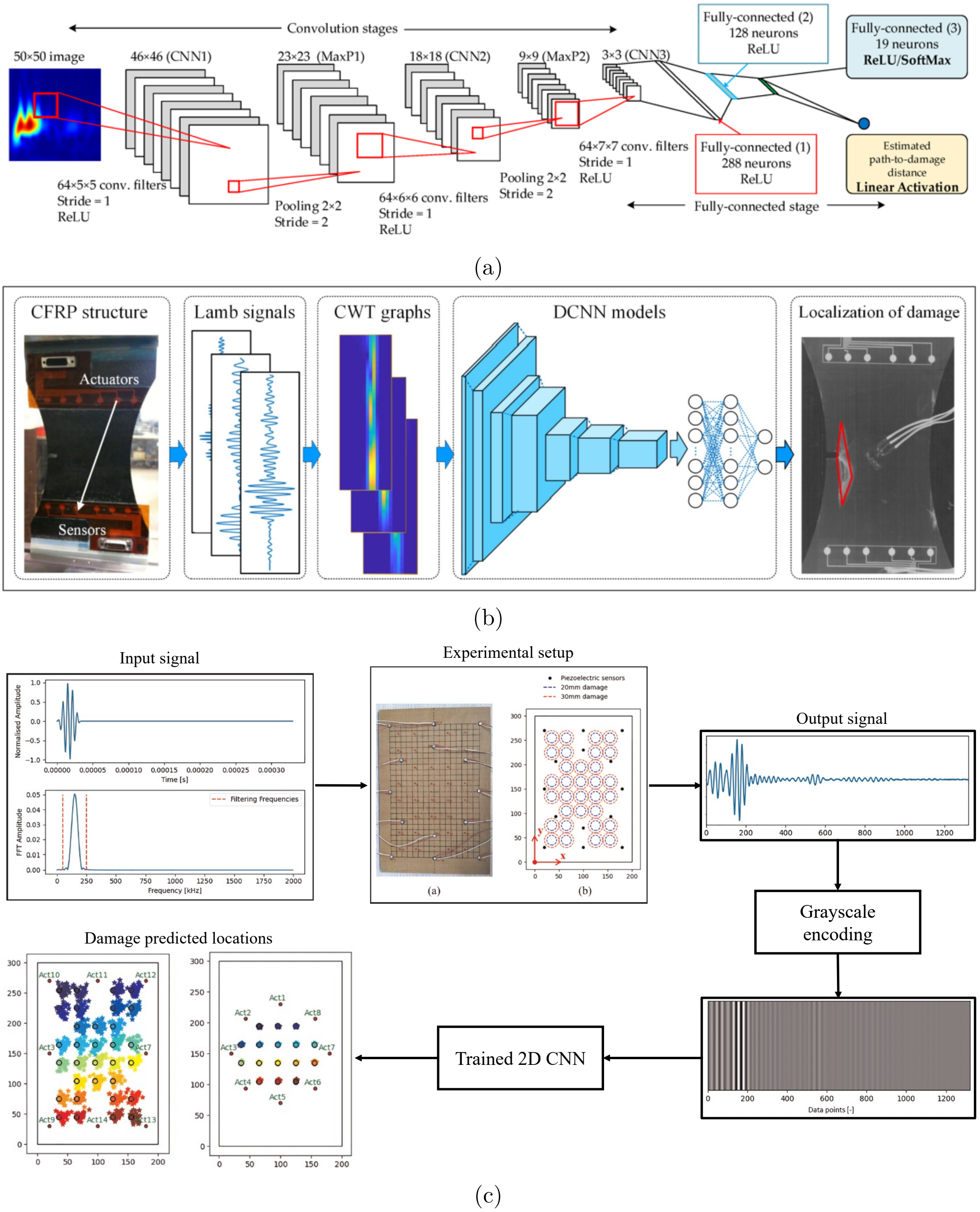
For composite structures, damage detection has been done using 2D CNNs. Azuara, Ruiz and Barrera (2021) and Wu *et al.* (2021) both propose 2D CNNs trained with time-frequency images generated with a continuous wavelet transform (CWT) applied to Lamb wave signals. The former uses this approach to predict simulated damage size, and the latter perdicts delamination area in plates obtained from the dataset at the Prognostics

Center of Excellence at NASA Ames Research Center. Gonzalez-Jimenez *et al.* (2023) also proposes a 2D CNN-based algorithm that localizes damage by processing Lamb waves. The authors used a Grayscale encoding method to transform the time series into grayscale images. Figure 15 presents the strategy used by the three studies. The main disadvantage of these approaches is the need of a pre-processing step, that requires computational effort and can lose information contained in the time series.

To minimize pre-processing steps, recent works used 1D CNNs, as these networks can process 1D arrays. The work of Ince *et al.* (2016) is arguably the first application of a 1D CNN for SHM purposes. The authors proposed a fast and accurate motor condition monitoring and early fault-detection system using 1D CNN and acceleration data. The proposed approach is directly applicable to the raw data (signal), and, thus, eliminates the need for a separate feature extraction algorithm resulting in more efficient systems in terms of both speed and hardware. Following this work, several other authors applied 1D CNNs as a damage detection algorithm for vibration data (AVCI *et al.*, 2017), structural joints (SHARMA; SEN, 2020), delamination (CRISTIANI *et al.*, 2022), and fatigue cracks detection (XU *et al.*, 2023), etc. Kiranyaz *et al.* (2021) presents an extensive discussion about 1D CNN and exemplifies this network structure with applications in various fields.

In the field of Lamb waves and metallic structures, Zhang, Li and Ye (2021) proposes a time-varying DI (TVDI) feature that preserves the temporal information to improve localization accuracy. This DI is calculated in a piecewise manner along the Lamb wave signal and the resulting vector is used as input to a 1D CNN, that localizes the damage. Shao *et al.* (2022) proposed a damage classification technique of aircraft aluminum plate-type structures using a 1D CNN. Pandey, Rai and Mitra (2022) employed the Local Interpretable Model-Agnostic Explanations with a 1D CNN to interpret the classifications of the model in terms of damage feature contributions.

Concerning composite materials inspection, only two works used 1D CNNs to process Lamb wave signals. Rai and Mitra (2022) proposed a 1D CNNs working as a feature extractor in a transfer learning strategy to detect and locate damage in composite structures. The strategy consists of two phases: on the first phase, they used a 1D CNN as an autoenconder (LI; PEI; LI, 2023), which was trained in a unsupervised scheme to learn key features of the Lamb wave signals. Thereafter, in the second phase, a classifier block is constructed stacking a 1D CNN layer, a fully connected layer and a binary classification layer (damage and undamaged classes). The autoencoder of the first phase is then imported and used as input to the classifier block. The system is then fine tuned with a training dataset and used to classify unseen damage. The authors



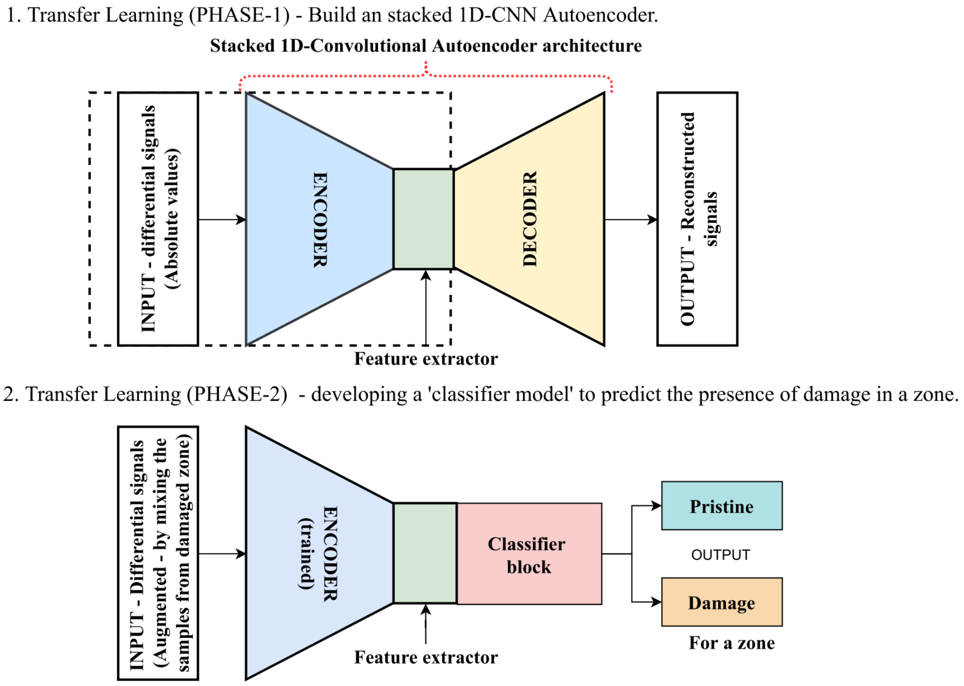
(a)

(b)

(c)

Source: Adapted from: (a) Azuara, Ruiz and Barrera (2021), (b) Wu *et al.* (2021), and (c) Gonzalez-Jimenez *et al.* (2023).

Figure 15 – Schematic representations of approaches using 2D CNNs: (a) methodology proposed by Azuara, Ruiz and Barrera (2021); (b) methodology proposed by Wu *et al.* (2021); and (c) methodology proposed by Gonzalez-Jimenez *et al.* (2023).


1. Transfer Learning (PHASE-1) - Build an stacked 1D-CNN Autoencoder.
Stacked 1D-Convolutional Autoencoder architecture
INPUT - differential signals (Absolute values)
ENCODER
DECODER
OUTPUT - Reconstructed signals
Feature extractor
2. Transfer Learning (PHASE-2) - developing a 'classifier model' to predict the presence of damage in a zone.
INPUT - Differential signals (Augmented - by mixing the samples from damaged zone)
ENCODER (trained)
Classifier block
Pristine
OUTPUT
Damage
For a zone
Feature extractor

Source: Rai and Mitra (2022).

Figure 16 – Flow diagram for the adopted transfer learning scheme involving a 1D CNN proposed by Rai and Mitra (2022).

achieved a classification accuracy of 97.91% during the training phase and 82.64% during the generalization test. Figure 16 depicts the transfer learning strategy.

Zhao *et al.* (2023) proposed a modified one-dimensional convolutional gated recurrent unit (MC1-DCGRU) for surface dent deformation and inter-laminated damage caused by low-velocity impact. The authors combined the feature extraction capabilities from 1D CNNs and the temporal information capturing ability of gated recurrent units (GRU). Their results indicate that this kind of model effectively captures damage features and temporal information from Lamb waves, enabling accurate damage localization with single-point excitation.

However, neither of these two authors explored the application of a 1D CNN directly on Lamb wave data for classification and quantification tasks. Rai and Mitra (2022) employed a 1D CNN as an autoencoder for feature extraction, and used the extraction as input for a separated classifier, while Zhao *et al.* (2023) utilized a wavelet denoising algorithm for signal pre-processing, and a modified 1D CNN+GRU for classification. Chapter 3 explores this research opportunity, using a 1D CNN directly on Lamb wave data with minimal pre-processing, and evaluates the performance of the algorithm in localizing damage compared with a classical MLP network fed with DIs calculated from the signal.

2.4 MODEL UPDATING

As presented in Section 2.3.3, neural networks require a substantial amount of data during the training process to effectively generalize over unseen data. In SHM applications, generating this volume of data exclusively through experimental means is not feasible. Therefore, numerical simulations can be employed to generate training data, and Section 2.2.3 discussed numerical techniques available for simulating Lamb wave propagation.

Nevertheless, for numerical models to accurately reproduce real structures, they need to be calibrated with experimental data. Ereiz, Duvnjak and Jimenez-Alonso (2022) provide an extensive discussion on finite element model updating methods. These methods can be categorized into two main areas: deterministic approaches and stochastic (or probabilistic) approaches. This Section introduces pertinent topics in the field of model updating, such as sensitivity analysis for reducing model complexity (2.4.1), surrogate modeling techniques (2.4.2), deterministic model updating methods (2.4.3), and stochastic model updating methods (2.4.4).

2.4.1 Sensitivity analysis

Sensitivity analysis evaluates the impact of variations in model parameters on the model's output (EREIZ; DUVNJAK; JIMENEZ-ALONSO, 2022). This process involves identifying metrics that represent the output and quantifying the influence of each parameter on these metrics. The complexity of sensitivity analysis escalates for nonlinear systems, where both individual parameters and their interactions contribute to the outcomes, particularly in multi-dimensional systems where numerous interactions must be assessed.

Consider the model response defined by:

$$Y = \mathcal{M}(\boldsymbol{\theta}), \tag{10}$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$ represents the vector of k input variables, and Y is the output scalar. In sensitivity analysis, the model inputs are considered as a random vector. To simplify, these inputs are assumed to be uniformly distributed, denoted by $\pi(\boldsymbol{\theta}_i) \sim \mathcal{U}(0,1)$.

Sensitivity analysis via Sobol indices involves decomposing the output Y into terms with increasing dimensionality, allowing for an orthogonal decomposition of the output:

$$Y = \mathcal{M}_0 + \sum_{i=1}^{k} \mathcal{M}_i(\theta_i) + \sum_{i<j}^{k} \mathcal{M}_{ij}(\theta_i, \theta_j) + \ldots + \mathcal{M}_{1\ldots k}(\theta_1, \ldots, \theta_k), \tag{11}$$

where $\mathcal{M}_0$ is the expected value of the output $E(Y)$. The terms in Eq. (11) are orthogonal, and each term can be calculated using conditional expectations. The terms $\mathcal{M}_i(\theta_i)$ and $\mathcal{M}_{ij}(\theta_i, \theta_j)$ are the conditional means for the parameters i and ij $(i \neq j)$ respectively (RAQUETI *et al.*, 2022):

$$\mathcal{M}_i(\theta_i) = \int_0^1 \dots \int_0^1 \mathcal{M}(\boldsymbol{\theta}) \, d\boldsymbol{\theta}_{\sim i} - \mathcal{M}_0, \tag{12}$$

$$\begin{aligned} \mathcal{M}_{ij}(\theta_i, \theta_j) = \int_0^1 \dots \int_0^1 \mathcal{M}(\boldsymbol{\theta}) \, d\boldsymbol{\theta}_{\sim ij} - \mathcal{M}_0 - \\ \mathcal{M}_i(\theta_i) - \mathcal{M}_j(\theta_j), \end{aligned} \tag{13}$$

where $\sim i$ indicates that parameter θ_i is held constant. The variance of the conditional expectation is used as a sensitivity measure. The first-order Sobol indices, indicating the effect of each input parameter on the output variance, are given by:

$$S_i = \frac{Var[\mathcal{M}_i(\theta_i)]}{Var[\mathcal{M}(\boldsymbol{\theta})]}. \tag{14}$$

These indices represent the main effect of each input parameter. The second-order indices, indicating the interaction effects, are:

$$S_{ij} = \frac{Var[\mathcal{M}_{ij}(\theta_i, \theta_j)]}{Var[\mathcal{M}(\boldsymbol{\theta})]}. \tag{15}$$

In this study, the first and second-order Sobol indices are computed to determine the parameters influence on the outputs of a FE model. This analysis is performed using Monte Carlo simulations and the UQLab framework (MARELLI; SUDRET, 2014).

2.4.2 Surrogate modeling

To compute Sobol' indices (Section 2.4.1) or perform Bayesian inference using MCMC (Section 2.4.4), it is necessary to perform multiple simulations. Although this task can be performed using the original FE model, the associated computational cost can be high and become computationally expensive. A possible alternative to address this problem is the use of surrogate models. The most popular surrogate modeling techniques include polynomial regression (LIU *et al.*, 2020; LIM; MANUEL, 2021), radial basis function (ZHOU *et al.*, 2016), Kriging predictor (WAN; REN, 2016), neural networks (PADIL; BAKHARY; HAO, 2017) and other methods.

In this work, two kinds of surrogate models are used: (i) polynomial chaos expansion, and (ii) neural networks. The former is used within the UQLab framework to compute Sobol indices in sensitivity analysis, and the latter is proposed as a novel approach to reduce the computational cost associated with simulating Lamb wave models in the Bayesian inference process.

Polynomial chaos expansion

The Polynomial Chaos Expansion (PCE) of a model response is a sum of orthogonal polynomials weighted by coeficients to be determined. They can be expressed as:

$$Y = \mathcal{M}(\boldsymbol{\theta}) = \sum_{\alpha=0}^{\infty} y_\alpha \Psi_\alpha(\boldsymbol{\theta}), \tag{16}$$

in which $\Psi_\alpha(\boldsymbol{\theta})$ represents multivariate orthonormal polynomials, and y_α are coefficients to be determined in order to perform the expansion. For computational implementation purposes, a finite version of the PCE is used

$$Y \approx \mathcal{M}^{PC}(\boldsymbol{\theta}) = \sum_{\alpha=0}^{P} y_\alpha \Psi_\alpha(\boldsymbol{\theta}), \tag{17}$$

where P is the number of terms in the PCE. The number of terms depends on the number of input random variables m and the maximum degree allowed for the polynomial expansion, p. According to Marelli and Sudret (2018) P is defined as:

$$P + 1 = \frac{(m+p)!}{m!p!}, \tag{18}$$

The choice of orthogonal polynomial family should align with the distribution of the model inputs. A detailed discussion on the families of polynomials and the methods for calculating their coefficients is available in Marelli and Sudret (2018). In this study, the UQLab framework (MARELLI; SUDRET, 2014) is utilized to construct a PCE based on input distributions. UQLab automatically selects the most suitable polynomial family according to the input data distribution.

ANN for surrogate modeling

As explained in Section 2.3.1, a neural network with one hidden layer sufficiently large can be proven to be an universal approximator. Therefore, these networks can be used as surrogate models given a proper training. Previous works have demonstrated the capacity of neural networks as surrogate models in applications such as general modeling (GORISSEN *et al.*, 2010), fluid flow (YETKIN; ABUHANIEH; YIGIT, 2024; GUO; LI;

IORIO, 2016), and solid mechanics (HAGHIGHAT *et al.*, 2021). Fakih et al. (FAKIH *et al.*, 2022) demonstrated the feasibility of neural networks as a surrogate model to simulate Lamb waves in aluminum structures.

In this work, a novel strategy for training and validating an ANN to act as a surrogate model for Lamb waves in a composite structure is presented. This surrogate model is used to speed up the Bayesian inference process. The complete framework is presented in Section 4.4.3.

2.4.3 Deterministic approaches for model updating

Deterministic updating is usually encoded as a minimization problem and aims at finding point estimates, or optimal values, for physical parameters. These problems can be solved with techniques ranging from traditional optimization algorithms, like the least-squares method, to more advanced ones like genetic algorithms (STANDOLI *et al.*, 2021) or particle swarm optimization(MARWALA, 2010).

Least squares

This work employs a modified version of the least-squares method for deterministic updating of model input parameters. The method's formulation, presented by Bud et al. (BUD *et al.*, 2022), is reproduced here for the reader's convenience. It consists of a multi-loss optimization technique that aims to minimize multiple loss function derived from the model outputs. To accomplish this, the influence of each input on the loss functions is linearized, and the numerical calculation of partial derivatives with respect to each parameter is conducted.

Assuming a calibration process that seeks a set of parameters $\boldsymbol{\theta}$ to minimize the discrepancy between a set of model outputs and experimental observations, the problem can be formulated as a constrained minimization problem. An error metric is defined as:

$$\varepsilon(\boldsymbol{\theta}) = \boldsymbol{Y_{target}} - \mathcal{M}(\boldsymbol{\theta}) \tag{19}$$

where $\boldsymbol{Y_{target}}$ represents the target values (e.g. experimental measurements) and $\mathcal{M}(\boldsymbol{\theta})$ denotes the outputs of the model $\mathcal{M}$ given the parameters $\boldsymbol{\theta}$. Note that multiple error metrics can be used.

Both qualitative and quantitative constraints can be applied to $\boldsymbol{\theta}$ to ensure the model's limits remain within physically plausible bounds. Quantitative constraints take the form of $(a_i \leq \theta_i \leq b_i)$, where a_i and b_i are the lower and upper bounds for the i-th variable, respectively. Qualitative constraints may include symmetry restrictions and relationships between not entirely independent parameters (e.g., the elastic moduli in plain weave composites typically have similar values and can have established relationships between them). This set of restrictions, combined with a model that may be underdetermined with respect to the number of equations relative to the number of variables, makes it impossible to obtain $\boldsymbol{\theta}$ in a single-step optimization. Therefore, an iterative approach is required, in which in each step, a set of initial values $\boldsymbol{\theta}_0$ is evaluated and the error between the model's output $\mathcal{M}(\boldsymbol{\theta}_0)$ and the target value $\boldsymbol{Y_{target}}$ is assessed.

The Taylor expansion of the error function in the vicinity of its initial value $\boldsymbol{\varepsilon}(\boldsymbol{\theta}_0)$, truncated after the linear term, and rearranged for minimization, is given by

$$\boldsymbol{\varepsilon}(\boldsymbol{\theta}) \approx \boldsymbol{\varepsilon}(\boldsymbol{\theta}_0) + \sum_i \left.\frac{\partial \boldsymbol{\varepsilon}}{\partial \theta_i}\right|_{\boldsymbol{\theta}_0} \Delta\theta_i \tag{20}$$

where $\Delta\theta_i$ represents small variations of the i-th parameter around the expansion point $\boldsymbol{\theta}_0$. The goal is to find the parameter increments $\Delta\theta_i$ that minimize the expanded error.

To minimize the error function, it is necessary to determine the variations $\Delta\theta_i$ in the calibration parameters that lead to a zero error function. The gradient components of the error function (ε) are derived by introducing incremental changes to each calibration parameter (θ_i), calculating the resulting incremental errors, and holding the other parameters $\theta_j, \forall j \neq i$, constant.

$$\left.\frac{\partial \varepsilon}{\partial \theta_i}\right|_{\theta_0} = \varepsilon(\theta_1, \theta_2, \ldots, \theta_i + 1, \ldots) - \epsilon(\theta_1, \theta_2, \ldots, \theta_i, \ldots) \tag{21}$$

By inserting Eq. 19 into Eq. 21, we obtain

$$\left.\frac{\partial \epsilon}{\partial \theta_i}\right|_{\theta_0} = -\mathcal{M}(\theta_1, \theta_2, \ldots, \theta_i + 1, \ldots) - \mathcal{M}(\theta_1, \theta_2, \ldots, \theta_i, \ldots) \tag{22}$$

which implies that the gradient of the error function can be calculated by the simple difference in the values of the results from the model models with parameters $(\theta_1, \theta_2, \ldots, \theta_i + 1, \ldots)$ and $(\theta_1, \theta_2, \ldots, \theta_i, \ldots)$.

Incorporating Eq. 19 into Eq. 20 results in a set of equations:

$$\sum_i \frac{\partial \epsilon}{\partial \theta_i}\bigg|_{\boldsymbol{\theta}_0} \Delta\theta_i = \mathcal{M}(\boldsymbol{\theta}_0) - Y_{target} \tag{23}$$

Eq. 23 usually defines an underdetermined system of equations. If different model outputs must have different significances, one can account for the varying significance by modifying the equations introducing a set of arbitrary weighting factors w_i, resulting in

$$\left[W \frac{\partial \epsilon}{\partial \theta_i}\bigg|_{\boldsymbol{\theta}_0} \right] \Delta\theta_i = W \left[\mathcal{M}(\boldsymbol{\theta}_0) - Y_{target} \right]. \tag{24}$$

In this context, W represents a diagonal matrix that contains the weighting parameters w_i. Consequently, each equation i in the system is scaled by a constant w_i, which corresponds to its significance within Eq. 23. Given that the system is under-constrained, there are multiple possible solutions. A higher weighting parameter will influence the system to prioritize the corresponding equation's observation more heavily.

This method is used in Chapter 4 to find optimal properties for the Lamb wave FE model, inside the proposed stochastic update framework.

2.4.4 Stochastic approaches for model updating

Unfortunately, any model is an idealization of reality and, therefore, incapable of representing its completeness. FE models often have significant uncertainty in their responses because of the lack of knowledge about material and geometric properties, as well as boundary and loading conditions. Stochastic approaches take these uncertainties into account during calibration steps, and include them in the updating procedure.

Among stochastic approaches, one of the most used is the Bayesian updating framework (SIMOEN; ROECK; LOMBAERT, 2015). This approach uses Bayesian inference to infer information about a posterior probability distribution of model parameters based on a prior probability distribution, experimental data, and a likelihood function. To generate samples of the unknown posterior, several methods have been proposed, such as Markov Chain Monte Carlo (MCMC), Transitional Markov Chain Monte Carlo (TMCMC), Sequential Monte Carlo sampler (LYE; CICIRELLO; PATELLI, 2021), Hamiltonian

Monte Carlo (BAISTHAKUR; CHAKRABORTY, 2020). Several works used Bayesian inference as a tool, including works with bolted joints (TELOLI *et al.*, 2021; MIGUEL; TELOLI; da Silva, 2022), vibration of beams with varying boundary conditions (RITTO; SAMPAIO; AGUIAR, 2016), and geometric non-linearities (WANG *et al.*, 2018).

Bayesian inference

Consider the scenario where model predictions, denoted as $\mathcal{D}^{\mathcal{M}}(\boldsymbol{\theta})$, are perturbed by an additive Gaussian noise with zero mean and variance denoted by σ_{ε}^{2}. This can be mathematically represented as:

$$\mathcal{D}^{\mathcal{M}}(\boldsymbol{\theta}) = \mathcal{M}(\boldsymbol{\theta}) + \epsilon, \tag{25}$$

where $\boldsymbol{\theta}$ is the vector of adjustable parameters, and ϵ encapsulates the errors stemming from measurement inaccuracies and uncertainties in model parameters.

The Bayesian approach seeks to identify the parameters $\boldsymbol{\theta}$ based on a set of empirical observations. By establishing fitting joint distribution $\pi(\boldsymbol{\theta})$, the model transitions into a stochastic form, capable of probabilistically forecasting the potential outcomes of the system's output $\mathcal{D}^{\mathcal{M}}$. A principal benefit of Bayesian inference in the context of model updating is its capacity to integrate prior knowledge with empirical data, thereby providing a stochastic representation of the inferred quantities (LYE; CICIRELLO; PATELLI, 2021).

Two foundational assumptions underpin this approach: (i) Given that the model parameters are treated as random variables, a prior distribution $\pi(\boldsymbol{\theta})$ is assigned to the input variables $\boldsymbol{\theta}$, reflecting pre-existing knowledge; (ii) Bayes' theorem is utilized to evolve the prior distribution into a posterior distribution, assimilating insights from random data observations denoted by $\mathcal{D}^{\mathcal{M}}$.

The posterior probability density function (PDF), $\pi(\boldsymbol{\theta}|\mathscr{D})$, is articulated as:

$$\pi(\boldsymbol{\theta}|\mathscr{D}) = \frac{\pi(\mathscr{D}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\mathscr{D})}, \tag{26}$$

where:

- $\mathscr{D}$ represents the vector of measurements or observations;
- $\pi(\boldsymbol{\theta})$ is the prior distribution;
- $\pi(\mathscr{D}|\boldsymbol{\theta})$ is the likelihood function for the parameters;

- $\pi(\mathscr{D})$ is the evidence or marginal likelihood; and
- $\pi(\boldsymbol{\theta}|\mathscr{D})$ is the posterior distribution.

- *Prior distribution*

The prior distribution, $\pi(\boldsymbol{\theta})$, encodes the pre-measurement knowledge about the parameters, which may be derived from expert judgment, smaller-scale experiments, or historical data. In line with Lye, Cicirello and Patelli (2021), when the parameters' bounds are known, a uniform distribution is typically adopted as the non-informative prior, following the Maximum Entropy principle.

- *Likelihood function*

The likelihood function, $\pi(\mathscr{D}|\boldsymbol{\theta})$, quantifies the congruence between the measurements, $\mathscr{D}$, and the model's outputs, $M(\boldsymbol{\theta})$. Assuming that the discrepancies between observations and model outputs are normally distributed with zero mean and specified variance, the likelihood function is naturally modeled as a normal distribution (TELOLI *et al.*, 2021).

- *Marginal likelihood*

The evidence function or marginal likelihood, $\pi(\mathscr{D})$, serves as a normalization factor, ensuring the posterior distribution's integration to one. As this term is constant and independent of the model parameters in Bayesian inference, the posterior is proportionally related to the product of the likelihood and the prior:

$$\pi(\boldsymbol{\theta}|\mathscr{D}) \propto \pi(\mathscr{D}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{27}$$

- *Posterior distribution*

The posterior distribution, $\pi(\boldsymbol{\theta}|\mathscr{D})$, embodies the refined distributions of the model parameters post-measurement, mirroring the enhanced understanding of the parameters informed by the data $\mathscr{D}$.

Direct sampling from the posterior is challenging with conventional Monte Carlo methods due to the implicit nature of the posterior distribution, which is only discernible upon evaluating the prior and likelihood functions. Consequently, generating samples from the posterior is not straightforward; instead, samples can only be drawn from the prior and likelihood. In high-dimensional spaces, evaluating all possible combinations of prior and likelihood through standard Monte Carlo simulations is impractical. To circumvent this, MCMC methods are employed to construct a Markov chain within the model parameters space $\boldsymbol{\theta}$ whose steady-state distribution is the posterior distribution of interest $\pi(\boldsymbol{\theta}|\mathscr{D})$ (LYE; CICIRELLO; PATELLI, 2021).

Markov-Chain Monte Carlo

The MCMC/Metropolis-Hastings algorithm is utilized to sample the posterior density function. A key strength of the MH algorithm is its ability to sample from any probability distribution, provided that the function proportional to its actual normalized density, i.e., the posterior distribution in the form of Eq. 27 is known and computable (LYE; CICIRELLO; PATELLI, 2021).

A Markov chain initiates from θ_1, and from there, it transitions between successive samples in the chain (i.e., from θ_i to θ_{i+1}). These candidates are sampled from a proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$. The acceptance of each candidate occurs with probability $\alpha = T(\theta_i \rightarrow \theta_{i+1})$. This probability is determined by the information gain from the current sample to the proposed sample, based on the Metropolis-Hastings acceptance criteria (CHIB; GREENBERG, 1995):

$$\alpha = \min\left(1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{D})}{\pi(\boldsymbol{\theta}|\boldsymbol{D})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right) \tag{28}$$

Considering $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ a symmetrical function (e.g. normal distribution) that is centered on θ_i, $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ $=q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$. Therefore:

$$\alpha = \min\left(1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{D})}{\pi(\boldsymbol{\theta}|\boldsymbol{D})}\right) \tag{29}$$

Assuming that the model predictions $\mathcal{D}^{\mathcal{M}}(\boldsymbol{\theta})$ are corrupted by an additive uncorrelated Gaussian noise ϵ (refer to Eq. 25) of zero mean and variance σ_ε^2, the analytical expression for the likelihood function is given by:

$$\pi(\mathcal{D}|\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\frac{\left(\mathcal{D}-\mathcal{D}^{\mathcal{M}}(\boldsymbol{\theta})\right)^T\left(\mathcal{D}-\mathcal{D}^{\mathcal{M}}(\boldsymbol{\theta})\right)}{\sigma_\varepsilon^2}\right). \tag{30}$$

And substituing the likelihood in Eq. 27, we have the Metropolis-Hastings acceptance criteria based on the current sample, the new sample and the experimental realizations.

Algorithm 3 presents the implementation of the MCMC/Metropolis-Hastings sampling algorithm. The random variables $\boldsymbol{\theta}$ are constrained to the interval $[\boldsymbol{\theta}_{\min}, \boldsymbol{\theta}_{\max}]$, with the current state normalized as $\boldsymbol{\theta}' = (1 - \boldsymbol{x})\boldsymbol{\theta}_{\min} + \boldsymbol{x}\boldsymbol{\theta}_{\max}$; here, $\boldsymbol{x}$ is a random variable uniformly distributed in $[0, 1]$ representing candidates generated for the posterior distribution. These candidates are sampled from a normal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ with a standard deviation of σ_p. Subsequently, the posterior distribution at the proposed sample are computed and the acceptance ratio is evaluated. Finally, the candidate is accepted or rejecetd based on the Metropolis-Hastings acceptance criteria. Figure 17 illustrates the random walk process graphically and showcase examples of acceptance and refusal of samples.

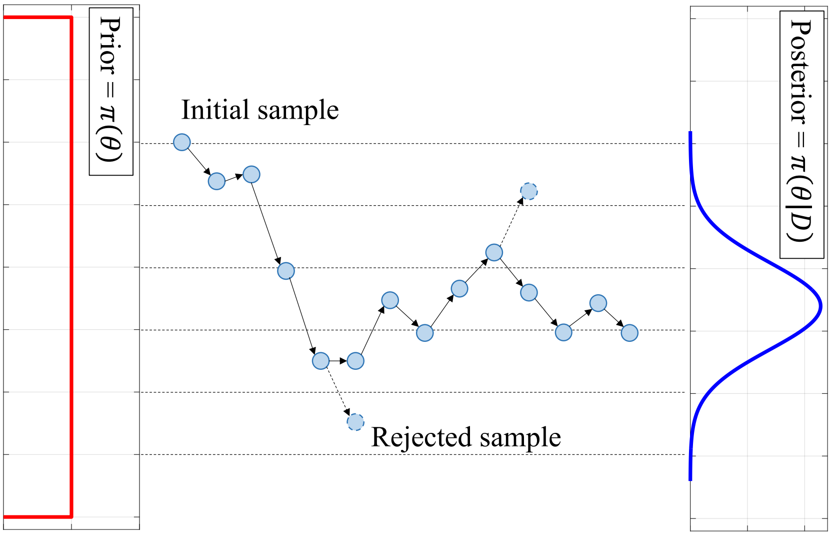
Algorithm 3: MCMC/Metropolis-Hastings Algorithm

```
Initialize $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_0$;
for $i \leftarrow 1$ to $N$ do
    Propose $\boldsymbol{\theta}^*$ drawn from $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$;
    Calculate acceptance ratio $\alpha = \min\left(1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{D})}{\pi(\boldsymbol{\theta}|\boldsymbol{D})}\right)$;
    Draw $u$ uniformly from $(0, 1)$;
    if $u < \alpha$ then
        Accept the proposal: set $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^*$;
    else
        Reject the proposal: keep $\boldsymbol{\theta}$ unchanged;
    end
    Save the current state $\boldsymbol{\theta}$ to the sample;
end
```

Source: Prepared by the author.

The hyperparameter σ_p is critical in controlling the random walk step size to prevent the chain from becoming static while allowing for adequate exploration of the parameter space. As suggested by Lye, Cicirello and Patelli (2021), adjusting σ_p to achieve an acceptance rate of candidates between 15% and 50% ensures that the efficiency of the algorithm remains at least 80%.

In this work, MCMC simulations are used in Chapters 4 and 5 to sample posterior distributions for material properties in a FE model for Lamb wave simulations.



Source: Prepared by the author.

Figure 17 – MCMC randon walk with visualization of the acceptance criteria.

2.4.5 Model updating applied to Lamb wave models

Several works in the literature use Lamb wave simulations (YANG *et al.*, 2006; MITRA; GOPALAKRISHNAN, 2016; DENG *et al.*, 2020). The majority of these works employ deterministic models with experimentally measured properties (SU; YE, 2005a; SU; YE, 2005b; PERFETTO *et al.*, 2022) or values updated using a deterministic updating approach (GRESIL *et al.*, 2013).

Few works in literature have used Bayesian inference to update Lamb wave models, especially for composite structures. Vanli and Jung (2014) updated unknown parameters of a 1D FE model and estimates a bias-correcting function to achieve a good match between the model predictions and sensor observations. Gallina *et al.* (2017) proposed a methodology for mechanical properties identification based on Bayesian inference and semi-empirical dispersion curves obtained with a LISA model. Yan *et al.* (2020) used multiple frequency excitation to obtain experimental dispersion curves and developed a scheme based on WFE method. Then, combining a Kriging predictor with Transitional Markov Chain Monte Carlo rounds, they sampled posterior PDF for laminate properties. All these studies share a common feature in the usage of specific Lamb wave simulation techniques such as LISA, the WFE method or a special 1D FE model. These methods need to be manually implemented as they are not readily available on commercial analysis software. Also, neither of these works investigated the effect of temperature on the mechanical properties or the Bayesian inference process. Chapter 4 presents the implementation of a framework based on Bayesian inference for updating a FE model to account for experimental uncertainties and tempreraturе variations.

2.5 STATE-OF-THE-ART OVERVIEW

This chapter provides an overview of the current state of research in SHM, focusing on four key areas: SHM practices, Lamb wave mechanics, neural network applications, and model updating methodologies. The recent surge in deep learning research applied to SHM showcases the scientific community's growing interest in leveraging these algorithms for damage detection tasks.

A notable research gap identified is the limited exploration of using unprocessed time series data for damage localization in composite structures via Lamb waves and 1D CNNs. While most studies have concentrated on data that has undergone some form of processing or transformation, the potential of usage of raw data with these algorithms remains an open field for research. Moreover, there is a lack of investigation into how the quantity of training data and the spatial discretization affect the precision of 1D CNNs in damage detection and localization. Addressing this research void could lead to the development of more effective and reliable SHM techniques that capitalize on the direct analysis of Lamb wave data using 1D CNNs. This topic is addressed in Chapter 3

Additionally, the integration of experimental uncertainty into the process of updating FE models is an area that has not been extensively covered in the literature. There is a promising avenue for creating dependable FE models that not only reflect experimental outcomes but also incorporate the inherent uncertainties of experimental data into the model parameters, resulting in a stochastic FE model. The dual advantages of such a model are significant: it can capture the range of experimental data within its confidence intervals, and its stochastic nature enables the generation of diverse simulations, each yielding distinct data realizations. These realizations can then be used to enrich the training datasets for machine learning algorithms, potentially enhancing their ability to generalize and predict unseen scenarios in SHM applications. The application of Bayesian inference in Lamb wave model updating and the usage of numerically generated data for training 1D CNNs is presented in Chapters 4 and 5.

3 DAMAGE LOCALIZATION USING LAMB WAVES AND 1D CONVOLUTIONAL NEURAL NETWORKS

The application of deep learning algorithms for analyzing Lamb wave signals represents a novel trend in signal processing, as elaborated in Section 2.3.3. Among these methods, CNNs have emerged as prominent tools for Lamb wave analysis (SU *et al.*, 2019b; ECKELS *et al.*, 2022; ZHANG *et al.*, 2022; RAMALHO *et al.*, 2023; GONZALEZ-JIMENEZ *et al.*, 2023), and in recent years, 1D CNNs (RAI; MITRA, 2021; SHAO *et al.*, 2022) have gained traction. These network architectures process one-dimensional arrays, such as time series, and hold promise for advanced signal processing applications.

However, current works in the literature that use 1D CNNs to analyze Lamb wave signals employ these algorithms with some type of pre-processing, e.g., wavelet transform (ZHANG; LI; YE, 2021), encoder-decoder schemes (RAI; MITRA, 2022), or combined with other neural network architectures, such as gated recurrent units (GRU) (ZHAO *et al.*, 2023). Consequently, this chapter explores a novel approach to localize damage in composite structures using Lamb waves and 1D CNNs based on processing directly Lamb wave time series. These time series undergo minimal pre-processing, only to remove high-frequency noise and to window relevant wave packets. The damage localization strategy is proposed in a global-local approach, with one 1D CNN being responsible for localizing the damage region and another for pinpointing the damage position.

The Chapter starts by introducing the Lamb wave detection (LaWaDe) system developed by the Acoustics and Vibrations Group at UFMG (Gravi UFMG). Section 3.1 describes the LaWaDe system and proposes an automation strategy to enhance its data collection efficiency. This automation is important for acquiring data under multiple damage conditions, essential for the effective training of machine learning algorithms. Manual collection of extensive signal data with the current system configuration is impractical.

Following this, Section 3.2 details the application of the system to gather data from multiple damage conditions in a composite material plate, as well as data pre-processing and storage routines. This Section also includes considerations about data quantity and availability.

Section 3.3 then delves into the use of 1D CNNs applied directly to Lamb wave time series for damage localization in the experimental setup, comparing the outcomes with those obtained from a traditional MLP network that uses Damage Indices (DIs) as input parameters. The performance of these networks is examined under three data availability scenarios in the training phase, simulating data constraints in real-world applications.

Thereafter, Section 3.4 provides comprehensive guidelines and recommendations for the development of 1D CNNs for Lamb wave signal analysis based on the author's experience during this research. Finally, Section 3.5 presents the concluding remarks.

The work presented in this chapter is based on the article "Damage localization on composite structures using Lamb waves and 1D convolutional neural networks", currently under review for publication in the Smart Materials and Structures journal.

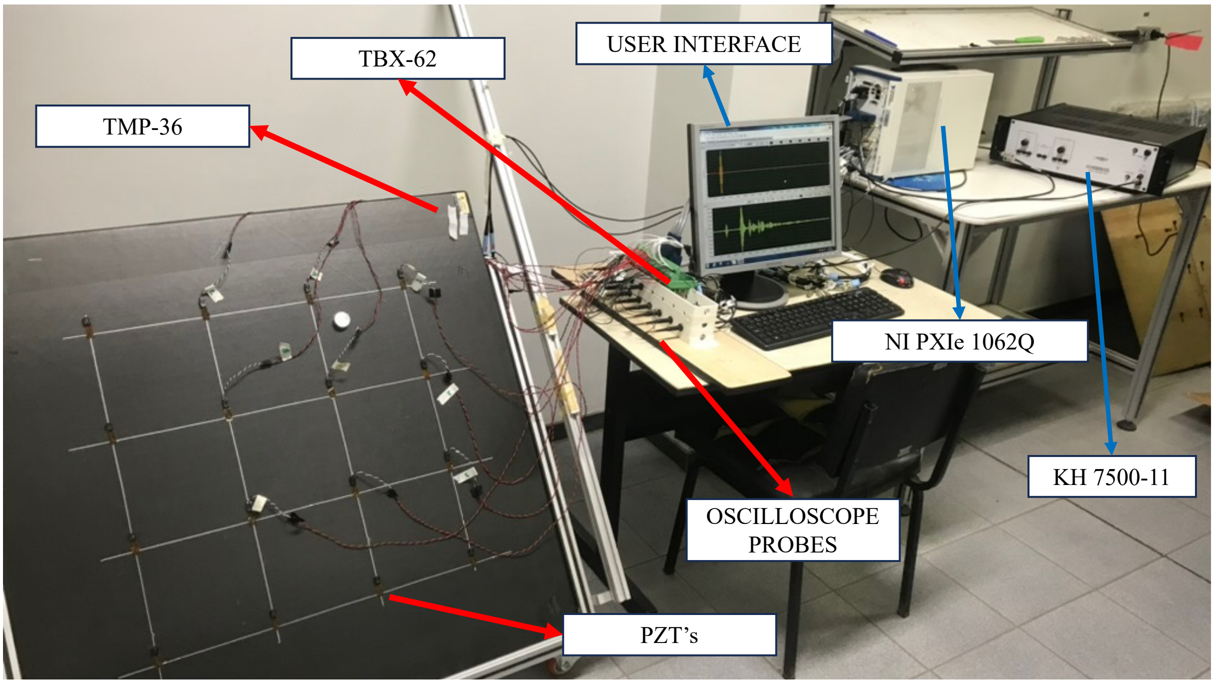
3.1 LaWaDe SYSTEM

As presented in Chapter 2, Lamb waves are a type of ultrasonic wave that propagates in thin-walled structures and can travel long distances. These waves interact with the medium in which they propagate, making them useful for damage inspection. Lamb waves can be generated in structures using PZT sensors and, in practical applications, a structure should be equipped with a sensor array, operating in a pitch-catch configuration to extend the coverage area (WANG *et al.*, 2020). The measured Lamb wave signals can be compared to a reference signal for damage assessment using various techniques, as presented in Section 2.2.4.

Lamb wave signal measurement can utilize multiple data acquisition strategies, including oscilloscopes (SILVA *et al.*, 2020), data acquisition boards (WANG *et al.*, 2020), or commercial systems such as the ScanGenie platform from Accelent Technologies. While well-suited for industrial applications, commercial systems are typically closed and rely on manufacturer-specific routines for post-processing and damage evaluation. However, for research purposes, a more adaptable and modifiable system is often preferred.

The GRAVi-UFMG, within the Department of Mechanical Engineering and in collaboration with the Department of Structural Engineering (DEES), developed a Lamb wave-based damage identification system named LaWaDe UFMG. This system was developed in a research project with a Brazilian aeronautical company and has contributed to conference publications(DONADON; FERREIRA; DUARTE, 2015; DONADON *et al.*, 2015), and a PhD. thesis (SANTANA, 2019). It is continually being improved, and there is a need to modify it to reduce the time necessary for measuring Lamb wave signals in multiple experimental campaigns.

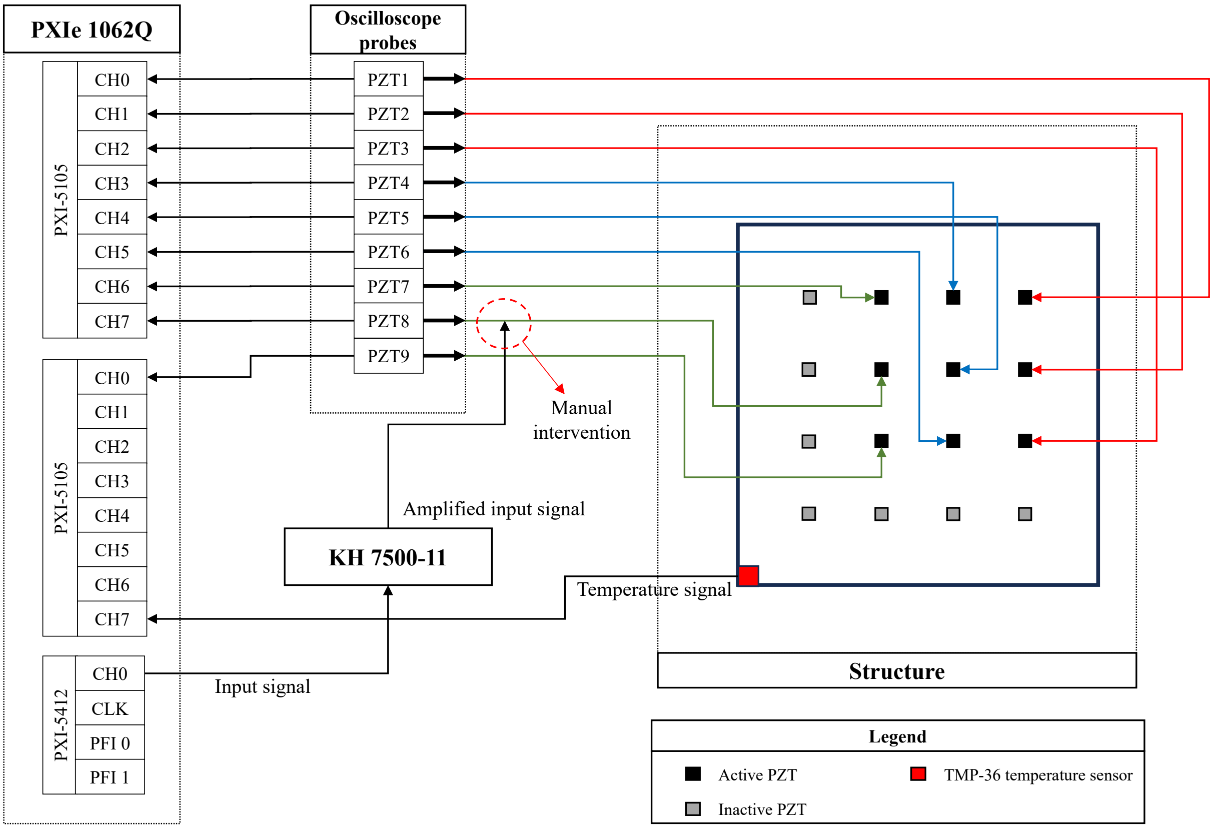
The LaWaDe system is built around a PXIe 1062Q acquisition platform from National Instruments. This platform integrates two NI PXI-5412 cards (14-Bit 100 MS/s) that are responsible for signal generation and two NI PXI-5105 acquisition cards (12-Bit 60 MS/s), each boasting eight channels, that are tasked with capturing the system's signals. Linked to the output of one of the NI PXI-5412 boards is a Krohn-Hite (KH) model 7500-11 amplifier. Its primary function is to amplify the input signal before directing it to the excitation PZT. Temperature readings are conducted using a TMP-36 sensor, which operates within a range of -40 to +125°C. Within this configuration, the system has one excitation source and up to 16 measurement channels. As one of the channels is connected to the TMP-36 sensor to capture temperature during the tests, 15 Lamb wave signals can be acquired simultaneously. Figure 18 presents the system with its main components.



Source: Prepared by the author.

Figure 18 – LaWaDe main hardware components.

Before initiating this doctoral research, the system operated on a manual strategy. This approach necessitated user intervention for each signal emission, specifically for



Source: Prepared by the author.

Figure 19 – Previous LaWaDe system wiring. Different wire colors are used to connect the oscilloscope probes to the structure, primarily to facilitate visualization. Note that the connection of the input signal had to be manually modified by the user.

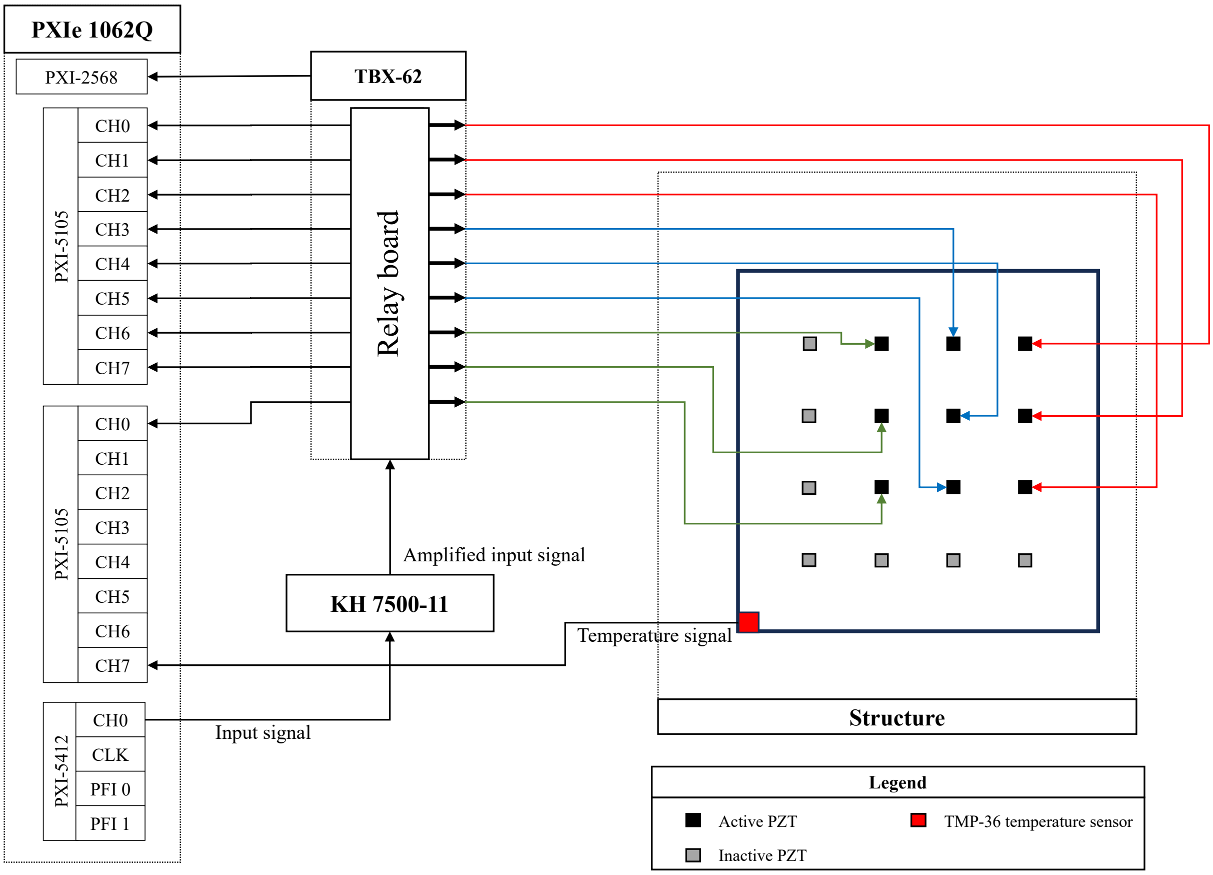
altering the excitation channel. Figure 19 illustrates the system’s wiring, featuring a 9 PZT array with temperature measurement capabilities. PZTs numbered 1 to 8 are connected to channels 0 through 7 on the first PXI-5105 board. Channel 9 connected to CH0 on the second PXI-5105 board, with temperature data recorded by CH7 on this board. Each PZT sensor interfaces with the PXIe 1062Q via oscilloscope probes offering variable attenuation options (1x and 10x). The excitation signal originated from the PXI-5412 card is amplified by the KH 7500-11 amplifier, and manually routed to a selected oscilloscope probe. To prevent saturation in the measurement channel, the attenuation for this probe is set to 10x. As shown in Figure 19, the excitation, for example, is connected to channel 8. This setup requires two manual adjustments whenever the excitation PZT changed: switching the excitation signal to a different channel and modifying the attenuation settings—reducing the previous excitation channel to 1x and increasing the new channel to 10x.

This configuration is both time-intensive and susceptible to operator error. A typical error involves the operator changing the excitation point to the intended oscilloscope probe, but forgetting to adjust the attenuation of either the original or the current excitation channel. Such an oversight results in either saturation of the excitation channel or excessive attenuation of one of the measurement channels, invalidating the collected data and requiring the initiation of a new experimental round. Furthermore, the system requires a manual routine for file saving, which includes manual naming and organization of measurements.

The automation of the measurement process is crucial for collecting sufficient data on different damage conditions to effectively train a machine learning, as manually collecting numerous signals with the current LaWaDe setup is impractical. Transitioning to automated operation of the system requires a key hardware feature: the capability for autonomous switching of excitation PZTs. The automation of the LaWaDe system was implemented through the integration of the TBX-62 relay board, interfaced with a NI PXI-2568 board. The TBX-62 is an interface featuring 31 relays, controllable via LabView. This board provides flexible connectivity between the PZTs, the amplifier, and the PXI system. Instead of manually switching between excitation and measurement by interchanging probe tips among the cables, the introduction of the relay board allows for all switching to be automated. It controls the activation and deactivation of the relays, thereby determining the operational role of each PZT, either as an exciter or a sensor. Figure 20 depicts the system interface with the presence of the TBX-62 and Appendix A presents an extensive discussion about the hardware implementation and wiring diagram.

The automation of the system has reduced the time required for a complete scan of the 9x9 array with all PZTs operating as sensors and actuators using 10 averages from 40 minutes to just 8 minutes. Now, the main time limitation of the system lies in the writing speed of the PXIe hard drive. Additionally, once the test begins, there is no need for user intervention until all channels have been automatically excited and the signals have been recorded. The signal recording process has also been automated in a data-saving routine internal to LabView.

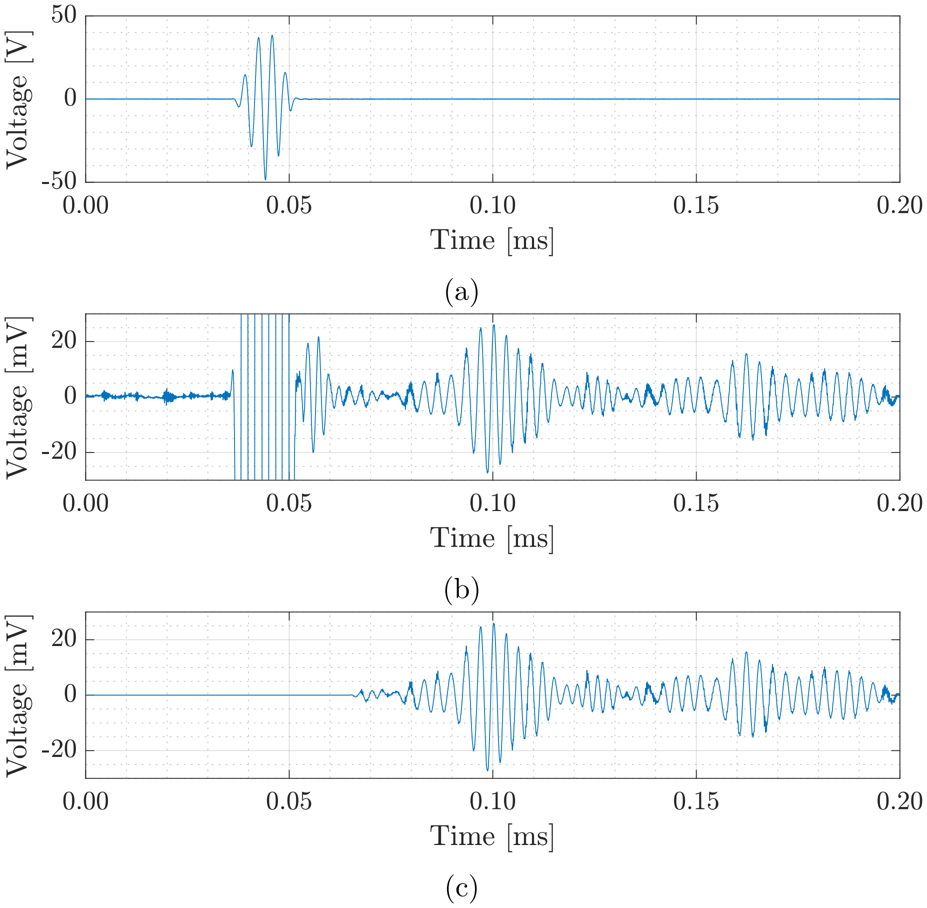
While the introduction of the relay board facilitates automatic channel switching and reduces test time, it has inadvertently increased crosstalk among the measurement channels. This issue primarily stems from the relay board's internal wiring being closely spaced. The excitation signal's amplitude ranges from 40 to 100 V post amplification, depending on the frequency under analysis. In contrast, the signals recorded from the piezoelectric sensors are in the 30 mV range. The internal cable proximity within the TBX-62 results in significant electromagnetic interference in the system when the excitation signal is active, as evident from Fig. 21. This figure showcases the input and output



Source: Prepared by the author.

Figure 20 – Automated relay connection implemented with TBX-62. Different wire colors are used to connect the oscilloscope probes to the structure, primarily to facilitate visualization.

signals in a test using a 5-cycle excitation packet at 300 kHz (see Fig. 21a). While the excitation signal remains active, the measurement channel, as seen in Fig. 21b, captures a similar signal that causes it to become saturated. Once the excitation signal concludes, the crosstalk ceases, allowing the measurement to proceed without interruption. To minize the saturation region, the input signal amplitude was limited to 50V, which slightly deteriorated the signal to noise ratio of the system. However, the crosstalk could be removed in a post-processing step, as demonstrated in Fig. 21c, and this removal does not affect the rest of the signal.



(a)

(b)

(c)

Source: Prepared by the author.

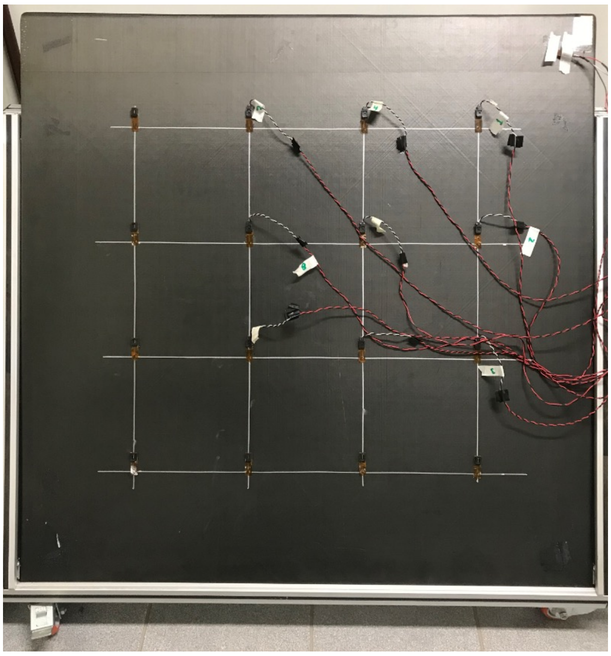
Figure 21 – Demonstration of crosstalk influence in measured signals after the adition of TBX-62 relay board: (a) input signal; (b) measured signal; and (c) measured signal with crosstalk removed in post-processing.

3.2 CASE STUDY

To evaluate the performance of 1D CNNs in tasks of damage localization, the automated version of the LaWaDe system is utilized to construct a comprehensive dataset featuring multiple damage positions within a composite structure. This Section outlines the structure under investigation, the methodology for damage data collection, the signal processing techniques applied to prepare Lamb wave data for input into the 1D CNNs, and the DIs employed as input for the MLP networks.

3.2.1 Structure

The study utilizes a carbon-epoxy laminate. This structure comprises a laminate of unidirectional carbon fabric and epoxy resin, featuring 16 layers with a stacking sequence of $[0/45/-45/90]_{2S}$. Each layer has a thickness of 0.19mm, and the plate dimensions are (1000 x 1000 x 3.04) mm. This laminate is selected due to the simplicity it offers for an initial analysis on a flat structure, as well as the availability of information and specimens in the laboratory.



Source: Prepared by the author.

Figure 22 – Carbon plate used on experimental campaign.

The laminate under investigation is equipped with a grid of 16 PZT sensors, spaced 200 mm apart both from each other and from the edges of the plate, both horizontally and vertically. In the analyses, only 9 sensors arranged in a 3x3 grid are utilized due to the 15 input channel limitations of the PXIe system. Moreover, this configuration represents a module that can be replicated for larger structures, thus generalizing the developed method.

Within this structure, damage is simulated using a 1.25-inch aluminum mass adhered to the plate. As indicated by Ihn and Chang (2008), concentrated masses can be used to simulate the localized alterations brought about by delamination in composite material structures. The mass is adhered to the plate using a polyisobutene-based adhesive named Pritt multitack.

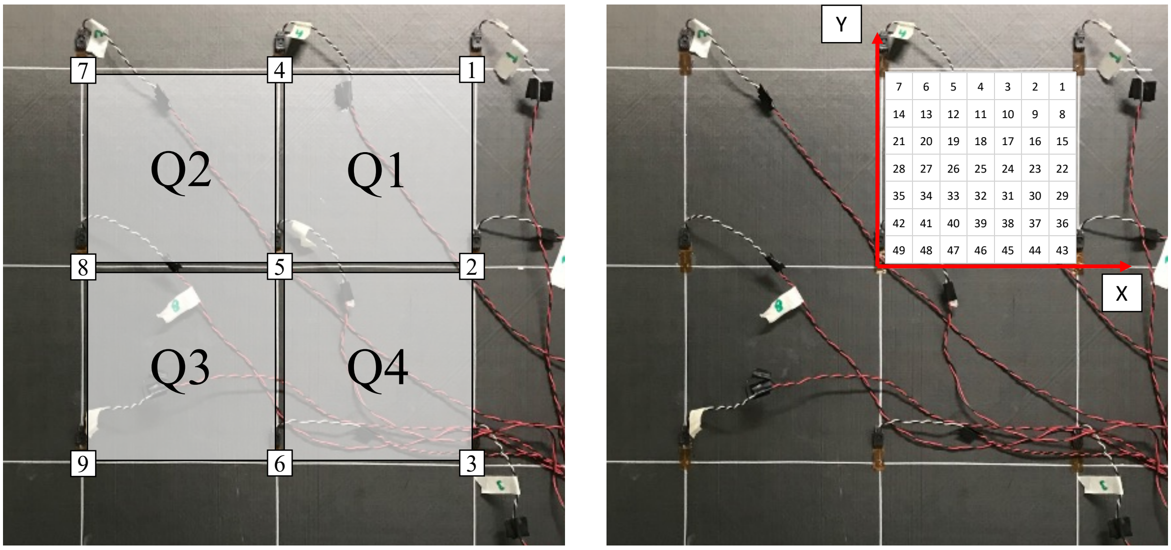
Given the dimensions of the plate, it is infeasible to use the thermal camera present in the mechanical enginnering department to ensure consistent temperature maintenance throughout the experiment. Instead, a temperature compensation strategy is implemented utilizing the OBS method. Temperature readings are systematically obtained for all experiments and a signature is captured at intervals of every ten tests. This interval means that a signal can deviate from a signature for up to five tests, equating to roughly 50 minutes. If the measured temperature deviates more than 1 °C from the current baseline, a new baseline with the nearest temperature is selected during post-processing. To further enhance the consistency of the testing environment, the room's ambient temperature is regulated by an air conditioner preset at 25°C during all experiments.

3.2.2 Experimental data acquisiton and processing

Data aquisition strategy

The upper-right region of the plate is the focus of this study, as illustrated in Fig. 22. The PZTs located in this region are sequentially numbered from 1 to 9, as depicted in Fig. 23a. Damage detection is conceptualized in a global-local strategy designed for scalability. PZT 5 is designated as the origin of a Cartesian coordinate system, with the X-axis extending horizontally to the right and the Y-axis vertically upwards. The structure is segmented into four quadrants, labeled Q1 to Q4, in alignment with the conventional trigonometrical circle layout. The core idea is that the method can be generalized to larger meshes comprising multiple 3x3 PZT grids. Initially, a neural network is trained to assess whether damage is present within the quadrant regions in a multi-class classification problem. Subsequently, a second neural network determines the precise coordinates of the damage in a regression problem. Section 3.3.1 details the application of a 1D CNN for quadrant identification, and Section 3.3.2 describes the development of a 1D CNN for coordinate determination.

Damage is simulated within these quadrants, specifically at the 49 positions (constituting a 7x7 grid) illustrated in 23b. While only the points for Q1 are depicted, 49 damage positions are evaluated for each quadrant, resulting in a total of 196 damage locations. For each point on the grid, a total of 10 measurements are collected. The evaluation of the 7x7 damage grid can also be extended to different data availability scenarios. Figure



7	6	5	4	3	2	1
14	13	12	11	10	9	8
21	20	19	18	17	16	15
28	27	26	25	24	23	22
35	34	33	32	31	30	29
42	41	40	39	38	37	36
49	48	47	46	45	44	43

(a) (b)

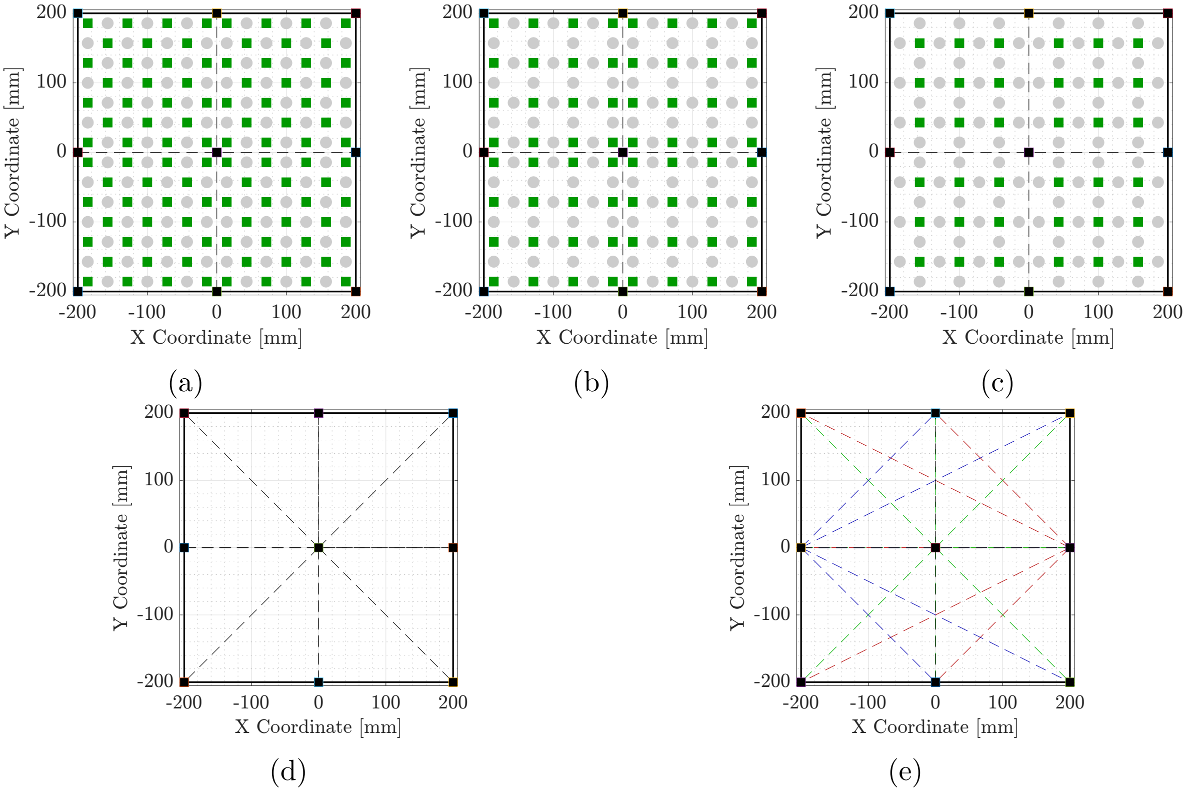
Source: Prepared by the author.

Figure 23 – Plate region under analysis and target labels for two stage localization: (a) Quadrant metric; (b) Coodinates for damage inside quadrant. Only coordinates for quadrant Q1 are shown, despite the same patern is present in Q2, Q3 and Q4.

24 depicts three distinct scenarios for training a neural network. In these scenarios, the 49 data points allocated to each quadrant are divided into 25 training points and 24 non-training points. The non-training points are further categorized into validation and test datasets. The efficacy of the neural networks on the non-training dataset is assessed using different quantities of training data. For instance, training sample sizes of 100, 64, and 36 can be utilized, corresponding to 25, 16, and 9 damage training positions per quadrant, respectively. Figures 24a, 24b and 24c depicts these scenarios.

Each channel can be used for excitation or measurement. For the networks evaluation, the selection of excitation and measurement PZTs also follows the global-local approach. During global inspections, a reduced signal method is employed. This method involves using solely the excitation signal from PZT 5, complemented by measurements from PZTs 1, 2, 3, 4, 6, 7, 8, and 9, as depicted in Fig. 24d. In contrast, the local analysis approach employs excitations from PZTs 2, 5, and 8 to maximize the coverage area (refer to Fig. 24e) while minimizing signal overlap and the input vector size.

The excitation signal is composed by a first part containing only zeros to observe the presence of vibration from the past signal, a second part containing five cycles of a 300 KHz sine wave multiplied by the hanning Windows with 50 V amplitude and a third part containing only zeros to assure that the structure vibration had been vanished. The observation window is defined as 0.4 ms with a sampling frequency of 60 MHz.

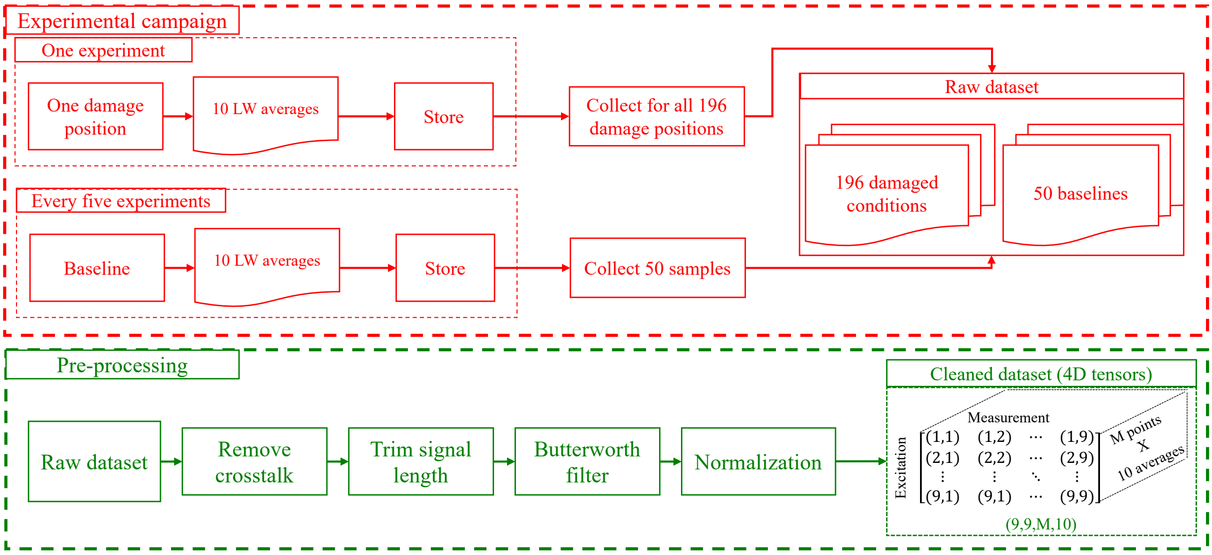


(a) (b) (c)

(d) (e)

Source: Prepared by the author.

Figure 24 – Different training scenarios used for ANN evaluation and excitaion signal paths for PZT array (a) 25 training samples per quadrant; (b) 16 training samples per quadrant; and (c) 9 training samples per quadrant. In each diagram, (■) represents training positions and (•) non-training positions. (d) Excitation path using only PZT 5 during global analysis to determine the damage quadrant, and (e) excitation path using PZTs 2, 5 and 8 during local analysis to determine the damage coordinates



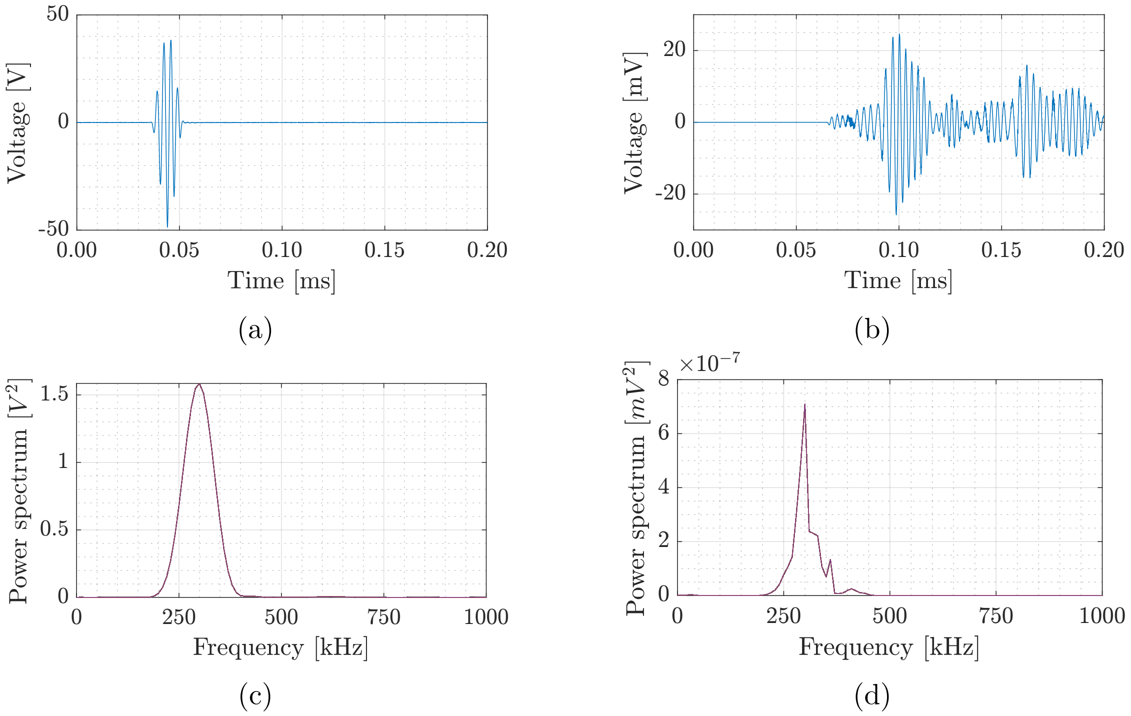
Source: Prepared by the author.

Figure 25 – Data acquisition and pre-processing strategies.

Data pre-processing

Figure 25 presents the steps for data acquisiton and pre-processing. During the experimental campaign, 10 measurements are collected for each damage position. These measurements are conducted using the pitch-catch configuration, in which one PZT acts as the emitter, and the remaining eight act as receivers at a time. Consequently, a signal grid of 9x9x10 is assembled, comprising 9 distinct excitation points, each with 9 measurements (including the excitation signal), and 10 averages. Additionally, after every ten tests, a set of 10 baseline signals are gathered. This pattern is repeated to accumulate data for all 196 damage positions, thereby constructing a comprehensive "raw signals" dataset (refer to the red section of Fig. 25).

The pre-processing phase comprises crosstalk removal, signal trimming and filtering, as shown in the green section of Fig. 25. The crosstalk region is removed from the measurement channels using the excitation channel signal as a reference (refer to Figs. 26a and 26b). Following this, the signal is trimmed into a 384 points window, aiming to preserve only the initial wave packets and ensuring focus on the most relevant content of the Lamb waves. A band-pass Butterworth filter with lower and upper cut-off frequencies set at 150 kHz and 450 kHz, respectively, is used to reduce measurement noise. These frequencies were selected to preserve the majority of the excitation signal's frequency content in Fig. 26c, and to ensure that no significant frequencies are lost, which is corroborated by the power spectral analysis of the receiver signals in Fig. 26d. Finally, all signals are normalized between -1 and 1 based on the peak amplitude of the baseline signals. The processed signals are then compiled into a “cleaned dataset” and stored in 4D tensors with dimensions $(9, 9, M, 10)$, in which $M = 384$ points in the time series. In total, there are 196 tensors for the damaged signals and 50 tensors for the baselines.



Source: Prepared by the author.

Figure 26 – Visualization of time and frequency contents of the input and output signal: (a) input signal on CH 5 in the time domain; (b) output signal on CH 1 in the time domain; (c) input singnal PSD; and (d) output signal PSD.

Neural networks input preparation

In the outlined experimental setup, a 3 × 3 PZT grid is used, comprising 9 potential excitation sources and an equal number of measurement points. For each excitation point, the excitation signal and the response in 8 sensors can be registered. Therefore, this configuration generates 81 potential Lamb wave signals for the machine learning algorithm to analyze. In a classical MLP, the amount of neurons and trainable parameters scale exponentially in this scenario. For example, using a single excitation point and eight measurement points, each capturing a signal with 1000 points, one training sample comprises 16,000 points, i.e., 8000 points for the baseline and 8000 points for the signal under evaluation. If these points are used as input for a first hidden layer comprising, for instance, 100 neurons, it results in a total of 1.6 million parameters. And that is only considering the first layer. Therefore, for the MLP network, a different strategy using DIs is used (refer to Section 3.2.3).

The primary benefit of a 1D CNN lies in its ability to process time series in a cost effective way. A 1D CNN can be constructed with three blocks of convolution, ReLU activation, and pooling to analyze the same data as the MLP, but having less than 1% of the parameters (refer to Table 1 for details).

Table 1 – Example of 1D CNN layers and dimensions.

Layer	Type	Hyperparameters	Size	Parameters

Input	Tensor input	-	8000x2	W = [3x2x4]
				B = [1x4]

Conv 1	1D Convolution	W = 3, S = 1 and F = 4	8000x4	-
ReLU 1	ReLU	-	8000x4	-
Avgpool 1	Average pooling	H = 16, andS = 16	500x4	-

Conv 2	1D Convolution	W = 3, S = 1 and F = 8	500x8	W = [3x4x8]
				B = [1x8]

ReLU 2	ReLU	-	500x8	-
Maxpool 1	Max pooling	H = 8 and S = 8	63x8	-

Conv 3	1D Convolution	W = 3, S = 1 and F = 16	63x16	W = [3x8x16]
				B = [1x16]

ReLU 3	ReLU	-	63x16	-
Maxpool 2	Max pooling	H = 4 and S = 4	16x16	-
Flatten 1	Flatten	-	256x1	-

FC 1	Fully-connected	N = 50	50x1	W = [50x16]
				B = [50x1]

ReLU 4	ReLU	-	50x1	-

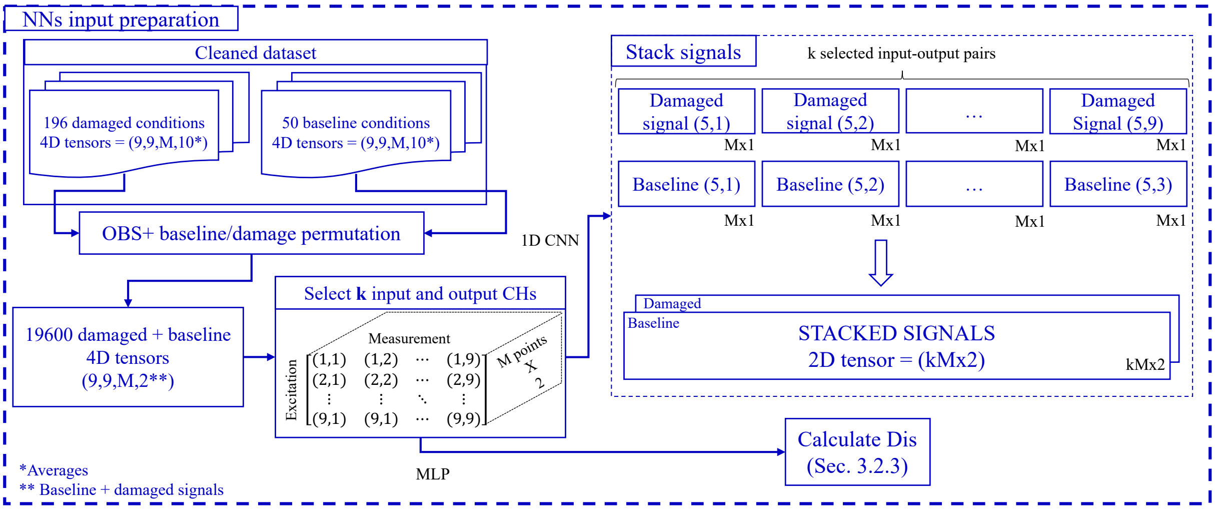
FC 2	Fully-connected	N = 1	1x1	W = [1x50]
				B = 1

Softmax 1	Softmax	-	1x1	-
TOTAL				13500 parameters

Source: Prepared by the author.

Figure 27 presents the data handling strategy for using the Lamb wave signals as input for the neural networks. The damaged conditions and baselines are combined through the OBS method. As each damaged condition and baseline is a tensor with 10 averages, i.e., the last dimension of the "Cleaned dataset 4D tensor" in Fig. 25, and these averages can be paired in a permutation strategy to create a total of 100 signal permutations. Therefore, the 196 damage positions result in 19,600 samples to be processed as inputs for the neural networks, in the form of a 4D tensor with dimensions $(9, 9, M, 2)$. Note that after the permutation, the last dimension represents the pair damaged signal and baseline.

To build the input for the neural networks, k input/output signals are selected from each tensor. In the global analysis, 8 signals pairs obtained with the excitation in PZT 5 are considered, as the excitation signal is not used as input for the neural networks. This corresponds to all signals $(5, 1, M, :)$ to $(5, 9, M, :)$ in the excitaion-measurement tensor of Fig. 27, except the signal $(5, 5, M, :)$. For the local analysis, signals $(2, 1, M, :)$, ... $(2, 9, :, M)$, $(5, 1, M, :)$, ... $(5, 9, M, :)$, and $(8, 1, M, :)$, ... $(8, 9, M, :)$ are selected, excluding



Source: Prepared by the author.

Figure 27 – Data preparation for using as input in the neural networks. The input-output pairs $(5,1)$ to $(5,9)$ are used to ilustrate the data selection in the global analysis.

$(2,2,M,:)$, $(5,5,M,:)$, and $(8,8,M,:)$. This represents 24 pairs of Lamb wave signals.

For the MLP network, three DIs are calculated for each pair damaged signal and baseline, as outlined in Section 3.2.3. Therefore, the MLP network processes 24 input parameters in a global analysis context and 72 parameters in a local analysis. The 1D CNN processes 2D tensors with dimensions $(kM, 2)$, with k representing the number of selected measurement PZTs and 2 representing baseline and damaged signals. These tensors are obtained by stacking the individual damaged and baseline signals into a 1D array, as depicted in Fig. 27. In the case of the global analysis, $k = 8$, and in the local analysis, $k = 24$, which results in tensors with $(3072, 2)$ and $(9216, 2)$, respectivelly. Finally, in both scenarios, the input parameters are correspondingly aligned with either the quadrant number (global analysis) or the normalized pair of coordinates (local analysis), with normalization ranging between 0 and 1.

3.2.3 Damage indices

A neural network used as an algorithm to interpret Lamb waves can accept two types of inputs: the direct Lamb wave time series or some metric or representation defined based on the signal, such as an image representing the wave pattern for a given instant or a representative metric of the signals, commonly referred to as Damage Index (DI). Employing DIs can reduce the dimensionality of the input data while also decreasing the required computational resources. However, as DIs converts the Lamb wave signals into one or more metrics, they lose spatial/temporal information contained in the signal.

Therefore, the handcrafted selection of the DIs is a challenge because they must be sensitive to the presence of damage. Dworakowski *et al.* (2015) evaluated the performance of MLP networks in damage localization and quantification using DIs as inputs, and three DIs proposed and validated by these authors are used in this work as input parameters for MLPs.

The first DI quantifies the vector difference between the Lamb wave signals :

$$DI_{\text{NORM}} = \frac{\displaystyle\int_{t_1}^{t_2} [y(t) - x(t)]^2 dt}{\displaystyle\int_{t_1}^{t_2} x(t)^2 dt}, \tag{31}$$

in which $y(t)$ and $x(t)$ are the measured signals with and without damage, respectively, and t_1 and t_2 are the integration intervals.

The second DI is calculated based on the cross-correlation of the signals. According to Dworakowski *et al.* (2015), the advantage of this indicator is that if the signal with damage and the baseline differs only in amplitude, the cross-correlation becomes the autocorrelation and has a maximum value of 1 at $\tau = 0$. This means that this DI is sensitive only to changes in shape and phase of the signal, not to changes in amplitude:

$$DI_{\text{XCOR}} = 1 - r_{xy}(\tau = 0), \tag{32}$$

in which $r_{xy}(\tau)$ is the cross-correlation operation.

The third DI used is based on the difference between the power spectral density of the damaged signal and the baseline, calculated as:

$$DI_{\text{PSD}} = \frac{\displaystyle\int_{f_1}^{f_2} [Y(f) - X(f)]^2 df}{\displaystyle\int_{f_1}^{f_2} X(f)^2 df}, \tag{33}$$

where $X(f)$ is the power spectral density of the baseline signal and $Y(f)$ is the power spectral density of the damaged signal.

3.3 RESULTS

This Section showcases damage quadrant classification and coordinate regression using the proposed strategy with 1D CNNs, comparing the results with those achieved using MLP networks fed by DIs, as mentioned in Section 3.2.2. The detection strategy employs a global-local approach:

- The detection of the damage quadrant is approached as a global analysis. The basic 3x3 structure can be indefinitely replicated in a larger structure, allowing for locating the damaged area. For this analysis, only the excitation at PZT 5 is utilized (refer to Fig. 24d).
- The detection of the damage coordinates is conducted as a local analysis. Once the quadrant is determined by a macro routine, the exact position can be locally assessed. This analysis uses excitations at PZTs 2, 5, and 8 (see Fig. 24e).

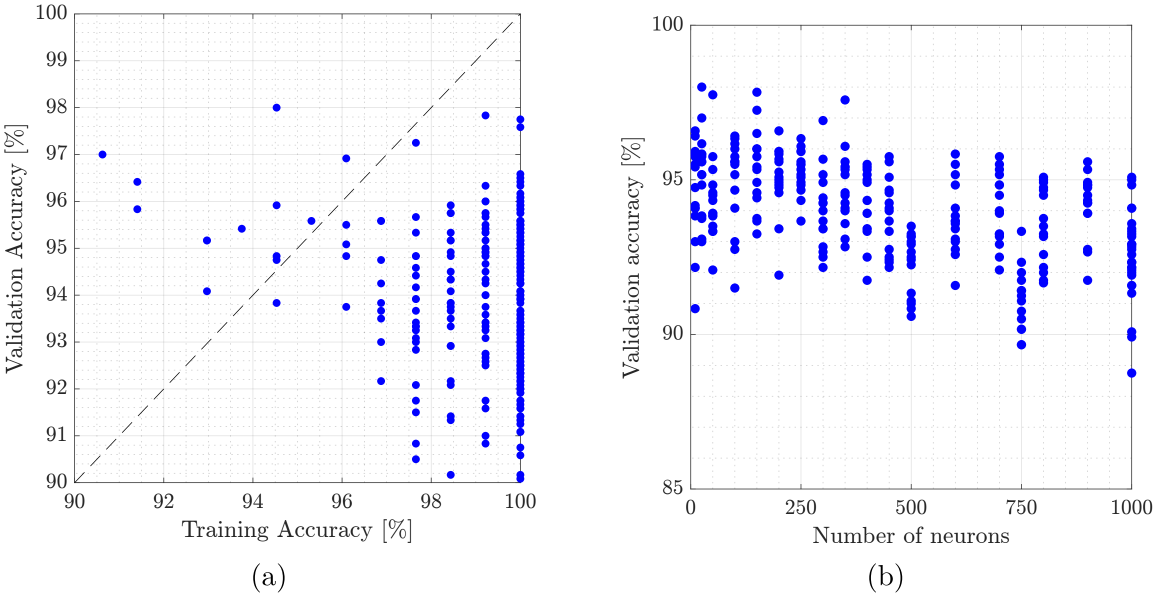
The following subsections detail the process involved in determining the network architecture for global and local damage localization. This is followed by an analysis of the results achieved in the context of damage localization. It is worth noting that the network architectures are designated using the naming convention outlined in Section 2.3.2, comprising I (Input), FC (Fully-Connected), C (Convolution), R (ReLU), P (Pooling), D (Dropout) and O (Output), in that order. Additionally, although not explicitly specified in the architecture diagrams, batch normalization layers are incorporated between convolution and ReLU layers. Their inclusion aims to enhance convergence during the training phase. Additionally, all neural networks are implemented using the deep learning toolbox of MATLAB. All training rounds are carried out in a computer MacBook Pro featuring a M2 Pro processor and 16 GB of shared memory.

3.3.1 Damage region localization

Network architecture selection

Damage region localization is a classification problem and the selection of network architectures is based on evaluating multiple configurations, and comparing them using the accuracy on validation data. In these simulations, the training data comprised 25 samples per quadrant to assess network performance under conditions of maximal data availability. This is supplemented by 12 random validation points in each quadrant, as shown in Figure 24.

For the MLP networks, configurations with two hidden layers and a dropout layer preceding the output layer, i.e., $(I - (FC - R) - (FC - R) - D - O)$, are examined. The neuron count in these networks ranged from 10 to 1000, and dropout probability from 10% to 50%. The MLPs are trained using DIs derived from the excitation signals at PZT 5 and readings from the other PZTs. The networks are trained using Adam as the optimization algorithm, a learning rate of 0.001, and an early stopping criteria based on the stagnation of the validation accuracy. Figure 28 presents the results for 5 networks sampled within the range for the proposed parameters. Note in Fig. 28a that there is a limit on validation accuracy in 98%. Multiple networks could be trained to reach 100% training accuracy, but at the risk of losing generalization capabilities.



Source: Prepared by the author.

Figure 28 – MLP performance during training phase for locating damage quadrant: (a) overfitting ratio as function of accuracy; and (b) accuracy in function of number of neurons.

One can see that there is a clear decrease in performance with increasing network size. Bigger networks can map more complex relationships in the training data, maximizing training accuracy, but at the cost of overfitting, i.e., loss of generalization capabilities. Therefore, the selection of the best candidate is based on the mean accuracy on validation data, and a network with 150 neurons and 10% dropout probability was selected.

Regarding 1D CNNs, the explored architectures vary from simpler two convolution-layer designs to more complex multi-layer configurations. These networks directly process Lamb wave signals from PZTs 1 to 9, with PZT 5 serving as the actuator. Architectures with three, four, and five convolution-ReLU-pooling blocks are simulated. However, most tested architectures are capable of dealing with the classification problem. This problem is relatively simple when compared to coordinate regression, as it consists of an outlier detection on the signals crossing a given quadrant. Therefore, among the multiple available architectures, the same one chosen for the regression problem is used to perform quadrant detection, consisting of the following structure: $I-(C-R-P)-D-(C-R-P)-(C-R-P)-FC-O$, as depicted in Fig 29. Section 3.3.2 provides a detailed discussion on the motivations behind this architecture choice and Fig.

29a illustrates the proposed architecture.

It is important to note that the only difference between the networks used for classification and regression is in the final layer, specifically the number of neurons and the activation function, as detailed in Section 2.3.2. The classification network includes a softmax function followed by four neurons, each representing the probability of damage presence in a specific quadrant, while the regression network calculates the half-mean-squared-error for each neuron, representing X an Y coordinates. Note that both networks use the same ‘‘feature extraction section’’.

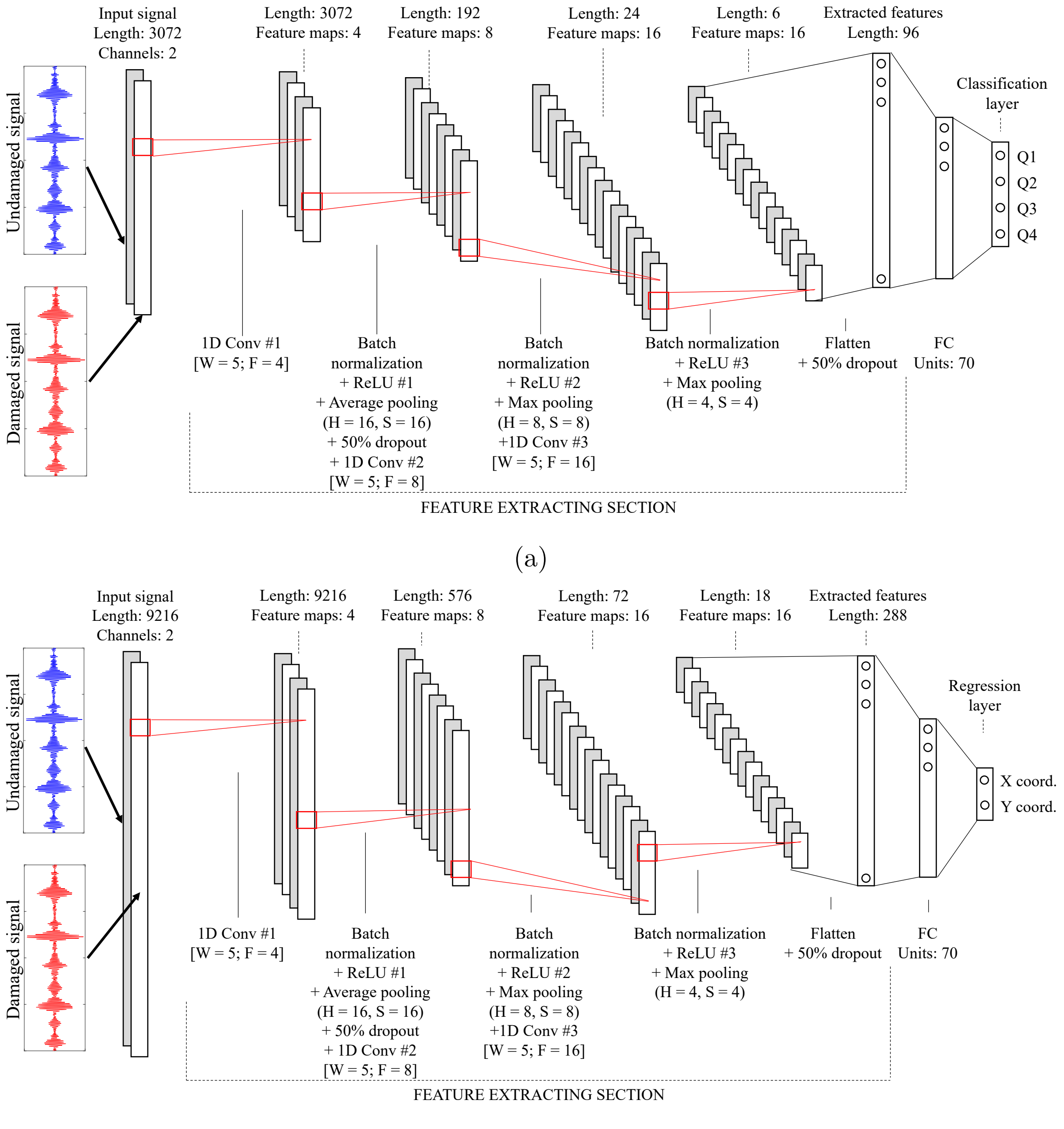
Quadrant localization results

The networks under consideration are assessed in training scenarios depicted in Figure 24. Ten networks are trained with each architecture and each amount of training samples per quadrant, i.e. 25, 16, and 9 samples. The networks are trained using the Adam optimization algorithm, a fixed learning rate of 0.001 and a stopping criteria based on the validation loss, i.e., the training process is stopped if validation loss do not decreases in three consecutive epochs.

The accuracy distribution for the each network architecture across the training, validation, and test datasets, as a function of the data volume used in the training phase, is illustrated in Figure 30. When trained with 25 samples per quadrant, the 1D CNN achieved a median accuracy of 100% on both validation and test datasets. In contrast, the MLP recorded median accuracies of 92.6% and 95.6% on the validation and test datasets, respectively. Notably, the MLP network exhibited a consistent accuracy of 100% on training data in all training scenarios, as shown in Figure 30a. This indicates a potential overfitting issue within the MLP network.

Both architectures exhibit a decline in performance correlating with the reduction of available data. This trend is evident in the downward trajectory observed in Figs. 30b and 30c. Specifically, when the training is conducted with 16 samples, the median test accuracy for the 1D CNN and MLP drops to 98.4% and 90.2%, respectively. A further decrease is noted when the training involves even fewer samples, with the median test accuracy reducing to 87.0% for the 1D CNN and 82.6% for the MLP. Additionally, it is important to note the difference in accuracy dispersion on test data between the two architectures; the 1D CNN demonstrates less variability compared to the MLP.

Figure 31 presents confusion charts for both MLP and CNN median networks, trained with various data volumes. With 25 samples per quadrant, the MLP network

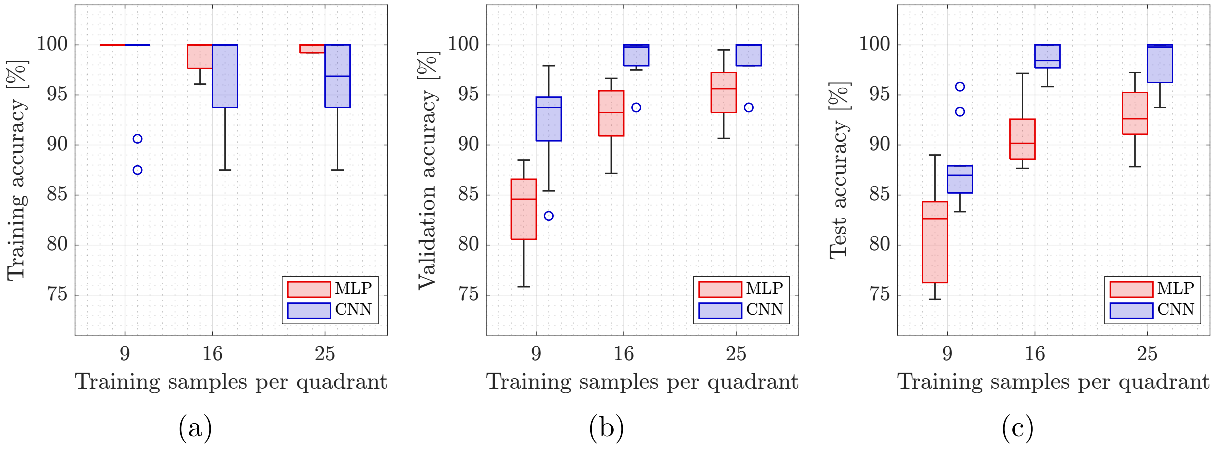


(a)

(b)

Source: Prepared by the author.

Figure 29 – 1D CNNs architectures selected for: (a) quadrant determination, and (b) damage localization. Note that is possible to use the same feature extracting architecture; modifying only the input and output layers

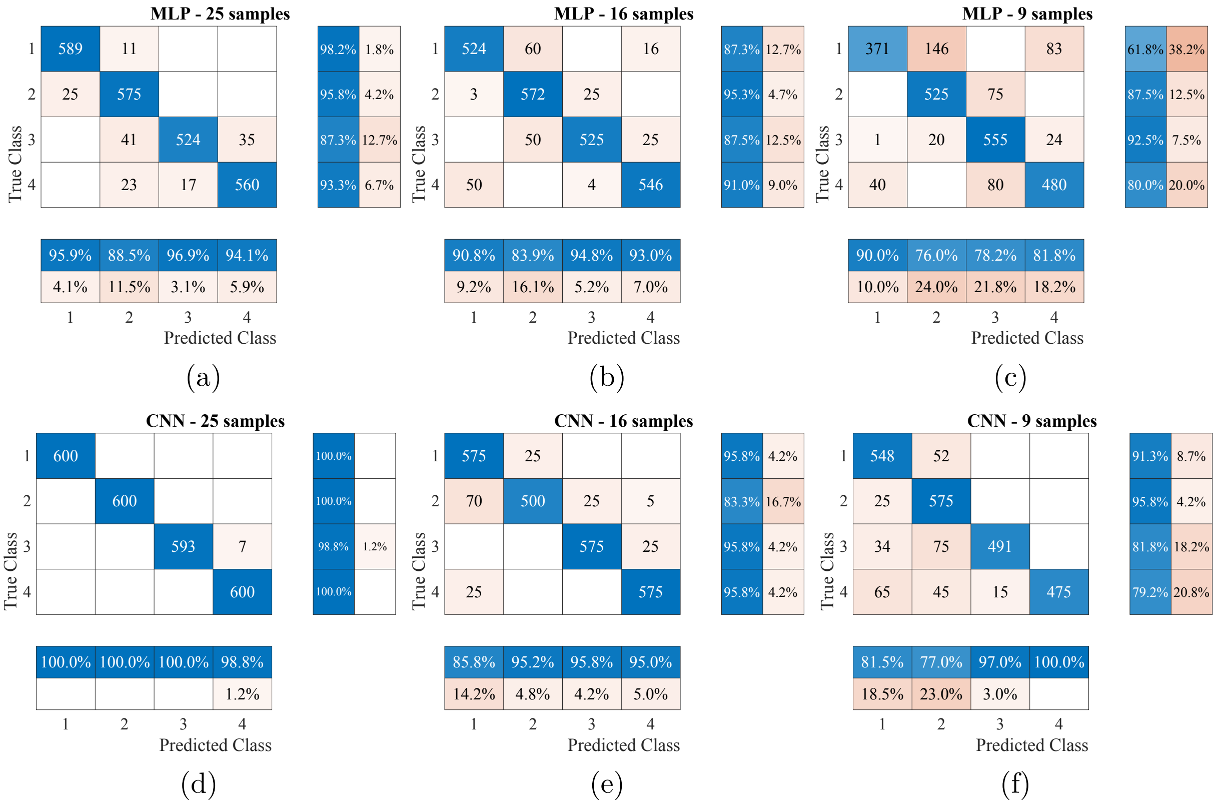


Source: Prepared by the author.

Figure 30 – Accuracy distribution with respect to the amount of data used in training stage: (a) training accuracy; (b) validation accuracy; and (c) test accuracy

performs satisfactorily, attaining accuracies ranging from 98.2% to 87.3% across individual quadrants. In contrast, the 1D CNN demonstrates superior performance, achieving 100% accuracy in test data for three of the four quadrants.

A reduction in training samples negatively impacts both networks, leading to an increased frequency of incorrectly predicted classes. For example, when the MLP network is trained with only 9 samples (as shown in Fig. 31c), its accuracy in predicting the first quadrant drops significantly to 61.8%, despite the median accuracy being around 82.6% as indicated in Fig. 30c. Conversely, the 1D CNN demonstrates greater resilience to data reduction, as can be seen in Fig. 31f. In a similar training scenario, the performance in the worst-predicted quadrant for the 1D CNN declines to 79.2%. This suggests that the poorest performance of the 1D CNN aligns closely with the median performance of the MLP.



(a) (b) (c)

(d) (e) (f)

Source: Prepared by the author.

Figure 31 – Confusion chart for test data on the median network of both architectures and different amount of samples per quadrant during training phase: (a) MLP using 25 training samples; (b) MLP using 16 training samples; (c) MLP using 9 training samples; (d) CNN using 25 training samples; (e) CNN using 16 training samples; and (f) CNN using 9 training samples.

3.3.2 Damage coordinates determination

Architecture determination

Localizing damage within a specific quadrant and accurately identifying the precise X and Y coordinates pose significant challenges, particularly in the context of designing effective network architectures for data analysis. This challenge is more pronounced in the case of 1D CNNs, given the wide array of architectural options available, which adds to the complexity of the task. For the architecture definition step, training data is defined as 25 samples, aiming to evaluate network performance under conditions of maximum data availability, complemented by 12 random validation points per quadrant (see Figure 24).

For MLP networks, simulations focus on structures with up to three hidden layers, with a uniform number of neurons across these layers to reduce the possible permutations and simplify the analysis. Networks with 10 to 500 neurons in hidden layers are examined. For 1D CNNs, the range of architectures spans from simple designs with two layers to more complex ones with several layers. This extensive evaluation is crucial to determine how different levels of network complexity influence the effectiveness of the system in accurately localizing damage. However, in this scenario, we also face the "Curse of Dimensionality", i.e., an increase in the size of the network consequentially increased the number of hyperparameters that must be considered and simulated in an exponential way. The definitions and hyperparameters of the CNNs includes: number of layers, organization of layers (e.g., convolution+ReLU+pooling, convolution+convolution,+ReLU+pooling, etc.), kernel size, number of kernels, stride, number of neurons for each FC layer, dropout probability for each dropout layer, etc. The variation in these parameters yields a multitude of configurations.

Initially, multiple configurations are evaluated, and Tab. 2 exemplifies the architecture of three MLPs and three 1D CNNs tested in this phase. The MLPs, labeled as #1, #2, and #3, feature one, two, and three hidden layers, respectively. CNN #1 and #2 are composed of three and four standard building blocks of convolution-ReLU-pooling, respectively. Meanwhile, CNN #3 incorporates a sequence of convolutions and activation functions before its initial pooling layer, drawing inspiration from renowned image recognition networks such as AlexNet, GoogleNet, and ResNet-50.

Table 3 lists the potential values for various hyperparameters, including kernel size in the convolution layers (W), number of filters in the convolution layers (F) and number of neurons in the last fully-connected layer (FC). The proposed architectures undergo

Table 2 – Example of three evaluated architectures. In this table, I, FC, C, R, P and O means input, fully-connected, convolution, ReLU, pooling and output, respectivelly.

Network	**Architecutre**
MLP #1	I-(FC-R)-D-O
MLP #2	I-(FC-R)-(FC-R)-D-O
MLP #3	I-(FC-R)-(FC-R)-(FC-R)-D-O
CNN #1	I-(C-R-P)-D-(C-R-P)-(C-R-P)-D-FC-R-O
CNN #2	I-(C-R-P)-D-(C-R-P)-(C-R-P)-(C-R-P)-D-FC-R-O
CNN #3	I-C-C-R-C-R-(C-R-P)-(C-R-P)-D-FC-R-O

Source: Prepared by the author.

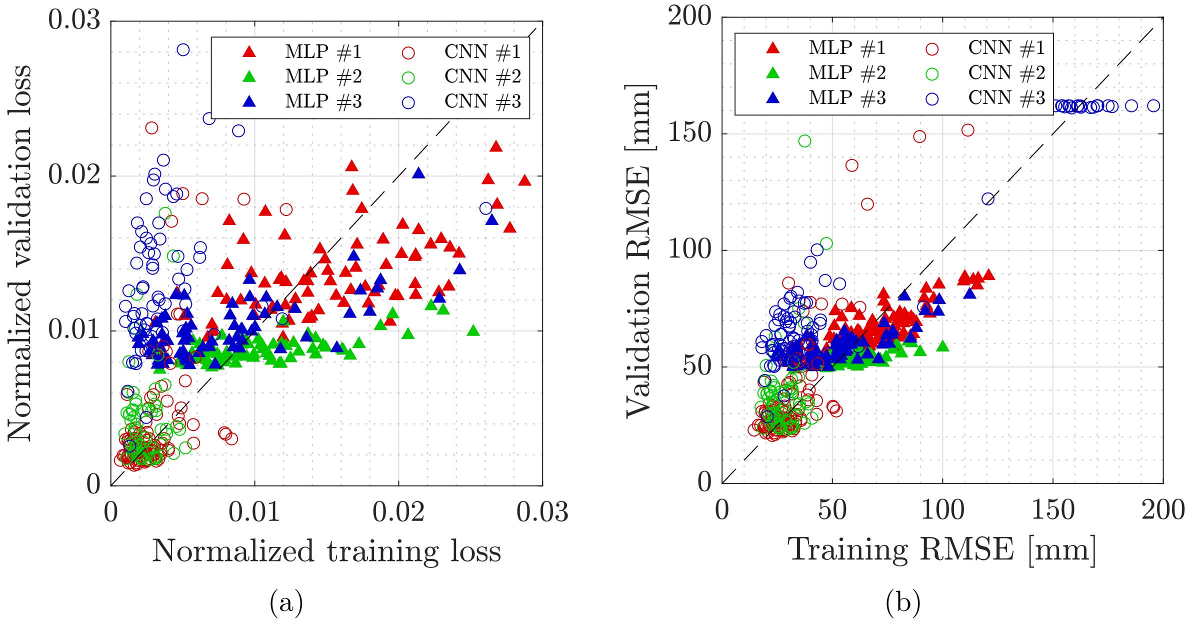
multiple simulations, up to five iterations depending on the convergence of results, to analyze the influence of initial hyperparameters and architecture on the outcomes. Figure 32 displays the loss on validation data as a function of the training loss, and the validation RMSE as a function of the training RMSE for some samples of the architectures in Tab. 2. Note that both graphs present the same general trend, as regression networks employ RMSE as their optimization loss. Each point represents a network with a set of hyperparameters, and they are grouped by architecture.

In these graphs, a lower training loss indicates a model's improved ability to fit the training data. Ideally, this should be paired with a reduction in validation loss, signifying the model's generalization capability. Networks with low training loss and high validation loss indicate overfitting, as shown in the upper-left part of the diagonal in Fig. 32a. Networks with high training loss suggest an inability to properly adjust to the data and are thus located in the right part of the graph.

Table 3 – Example of three evaluated architectures. In this table, W represents the kernel size of the convolution operation, F the number of channels per convolution layer, and FC the number of neurons in the last fully-connected layer.

Network	Parameters #1	Parameters #2	Parameters #3	Parameters #4
MLP #1	10 to 500 neurons	Dropout - 0 to 50%	-	-
MLP #2	10 to 500 neurons	Dropout - 0 to 50%	-	-
MLP #3	10 to 500 neurons	Dropout - 0 to 50%	-	-
CNN #1	W = [3, 5, 7, 9]	F = [4, 6, 8]	FC = [10 to 100]	Dropout - 0 to 50%
CNN #2	W = [3, 5, 7, 9]	F = [4, 6, 8]	FC = [10 to 100]	Dropout - 0 to 50%
CNN #3	W = [3, 5, 7, 9]	F = [4, 6, 8]	FC = [10 to 100]	Dropout - 0 to 50%

Source: Prepared by the author.



Source: Prepared by the author.

Figure 32 – Training and validation performance comparison between different MLP (▲) and 1D CNN architectures (∘) on validation data for: (a) loss; and (b) RMSE

When comparing the loss of both network sets, CNNs exhibit superior performance, achieving lower training errors while also reducing validation errors. Most results for CNNs #1 and #2 are in the lower-left part of the diagrams, demonstrating low error and a generalization capacity.

It is observed that some samples of MLP architectures #2 and #3 nearly match the best CNN networks in terms of normalized training loss. However, unlike the CNN networks, this reduced training loss does not translate to a lower validation loss in the MLP networks, suggesting overfitting. This occurs even with dropout probabilities of up to 50% in the final layer, implemented to enhance generalization. The MLP networks plateau at a validation loss of approximately 0.008 and a validation RMSE lower than 50 mm. This pattern indicates a lack of sufficient information on training data for the network to learn generalized patterns applicable to unseen data. Attempts to reduce validation loss with larger network architectures did not yield better results, implying a limitation likely due to information loss during the calculation of DIs. Consequently, for subsequent analyses, architecture MLP #2 is preferred over #3, as it offers comparable performance but is smaller and more efficient to train.

Generalization errors are also evident in the evaluation of CNN #3. Although several networks learn features from the training data effectively, resulting in low training errors, they exhibit high validation errors. This contrast in performance, particularly when compared to other CNN architectures that generalize well with the same training data, suggests that the larger networks, like CNN #3, are prone to overfitting due to their enhanced capability for representation. Additionally, one can see a concentration of samples of CNN #3 in the upper-right region of the Fig. 32b. These networks presented high error both for training and validation datasets. This indicates that the network became trapped in a local minima along the training process. This phenomenon is common for deep neural networks and is called *degradation*. In this phenomenon, with the network depth increasing, performance gets saturated and then degrades rapidly (HE *et al.*, 2015a). There are ways to counter degradation, e.g., the implementation of residual layers, but for the proposed application in this work, this is not needed, as simpler networks, such as CNN #1 and #2, are capable of dealing with the problem in question.

Several other architectures were simulated, and they fall in two main categories: (i) big networks (larger than CNN #2) and small networks (smaller that CNN #1). Bigger networks tend to be able to learn from training data, but often present overfitting, degradation or gradient vanishing problems. The gradient vanishing problem arises from the multiplication of small derivatives in deep networks during backpropagation, leading to extremely small gradients and hindering learning. Conversely, smaller networks struggle to effectively extract features and subsample the signal, resulting either in poor performance or networks with big flattened layers, dificulting training. To compensate, these networks often require a larger stride in the pooling layers. However, this approach risks discarding substantial information during the pooling process. Therefore, architecture CNN #2 is selected between the multiple tested because as it has fewer layers than CNN #3, meaning a simpler model with fewer hyperparameters and faster training phase.

With the definition of the architectures, the following phase is the hyperparameter optimization. For this phase, some constraints are imposed to limit the possible combinations. They are:

- Uniform kernel size across all convolution layers, mirroring practices in leading image analysis networks like VGG-19 and ResNet.
- A fixed stride equals to one in all convolution layers.
- Pooling layers designed to reduce the signal by powers of 2, such as 2, 4, 8, 16.
- A multiplicative increase in the number of filters in consecutive layers following a pooling layer. For instance, $\text{Filters}_{L2} = 2 \times \text{Filters}_{L1}$, $\text{Filters}_{L3} = 2 \times \text{Filters}_{L2}$, and so on.

More details and examples about these considerations are given in Section 3.4.

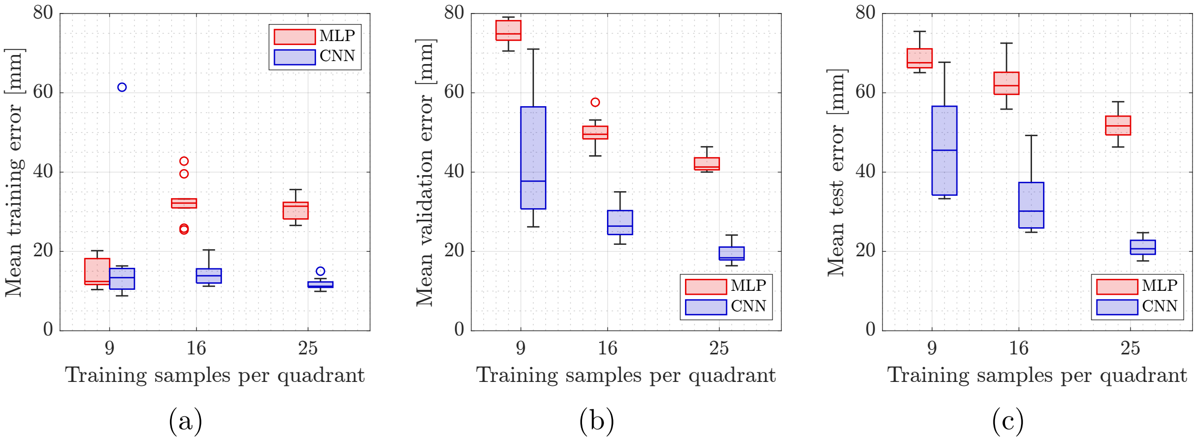
The problem of finding optimal hyperparameters becomes a search through four dimensions: kernel size, number of kernels in the first layer, number of neurons in the fully-connected layer and dropout probability. With these considerations, multiple networks with each hyperparameter configuration are evaluated. Table 4 presents the best 5 performing 1D CNNs obtained with architecture #1. Note that their normalized validation error is very similar, between 0.052 and 0.056. The normalization is performed using the 400 millimiters length of the scanning area (two quadrants) to speed up the training process. A metric named ‘‘overfitting ratio’’ is introduced, defined as the ratio between the validation loss and the training loss. A network with overfitting ratio greater than one indicates a degree of overfitting on training data. All networks present a good overfitting ratio, and the decision at this point is based on the author’s preference. Candidate is #1 selected by its lowest validation error, a overfitting ratio lower than 1 and three of the best five networks have kernel size equals five.

Table 4 – Five networks with the lowest validation RMSE

Network	**Kernel size**	**Number of kernels 1st layer**	**FC size**	**Dropout probability [%]**	**Normalized validation RMSE**	**Overfitting ratio**
Candidate #1	5	4	70	20	0.052	0.95
Candidate #2	7	4	50	20	0.053	0.88
Candidate #3	5	4	50	30	0.054	0.90
Candidate #4	5	6	70	40	0.055	1.06
Candidate #5	9	6	50	20	0.056	1.05

Source: Prepared by the author.

The same process is repeated for the MLP with architecture #2 and a network with 250 neurons and a dropout probability is selected.



Source: Prepared by the author.

Figure 33 – Error distribution with respect to the amount of data used in training stage: (a) training error; (b) validation error; and (c) test error

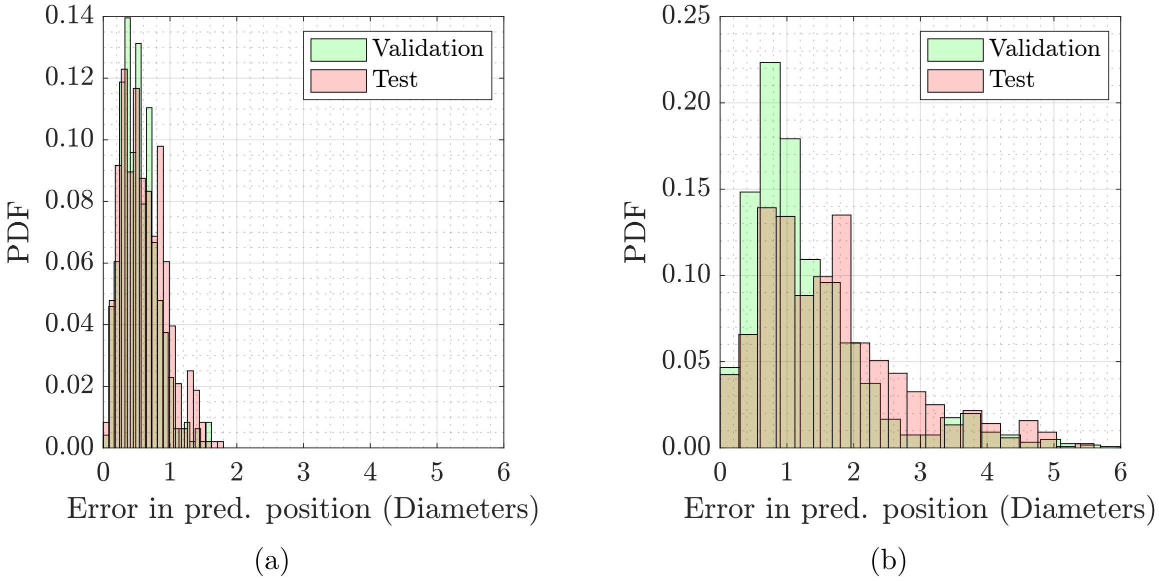
Coordinates determination results

Figure 29b presents the selected architecture for the 1D CNN for damage coordinates localization. This architecture, along with the selected MLP, is evaluated across different training scenarios as shown in Fig. 24. These scenarios include training with 25, 16, and 9 samples per quadrant, and ten networks being trained for each scenario of data availability.

Figure 33 displays the evolution of training, validation, and test errors for both the CNN and MLP architectures with respect to the amount of data used. Note that all networks achieved a lower training error than validation and test errors. However, the 1D CNN consistently outperforms the MLP in all training scenarios, and it has a lower validation and test errors than the MLP counterpart. Notably, the 1D CNN trained with all 25 samples per quadrant has a median validation error of 60% lower than the MLP network trained with the same amount of data. Additionally, the CNN trained with only 9 samples demonstrates superior performance compared to the MLP trained with 25 samples, presenting a median error of 45.5 mm compared to 51.7 mm. This advantage can be attributed to the CNN's capacity to process the entire Lamb wave signal during training, thereby extracting more relevant information than what is provided by the three DIs used in the MLP.

A reduction in training data leads to a decline in performance for all networks. Specifically, the MLPs exhibit an increase in mean error, though with minimal dispersion. Conversely, the 1D CNNs not only show an increase in error but also a greater dispersion of these errors. This trend is attributable to convergence issues common in deep learning algorithms when trained with limited data. Nevertheless, in this specific application, this challenge can be mitigated by monitoring the overfitting ratio of the networks during the training phase. The boxplots reveal varying patterns of validation errors among networks, with some displaying low and others high validation errors, yet most maintain low training errors. Implementing a strategy to automatically discard networks exhibiting high overfitting ratios during training would effectively address this issue. However, such a strategy is less applicable to MLPs due to the more concentrated nature of their errors around the median.

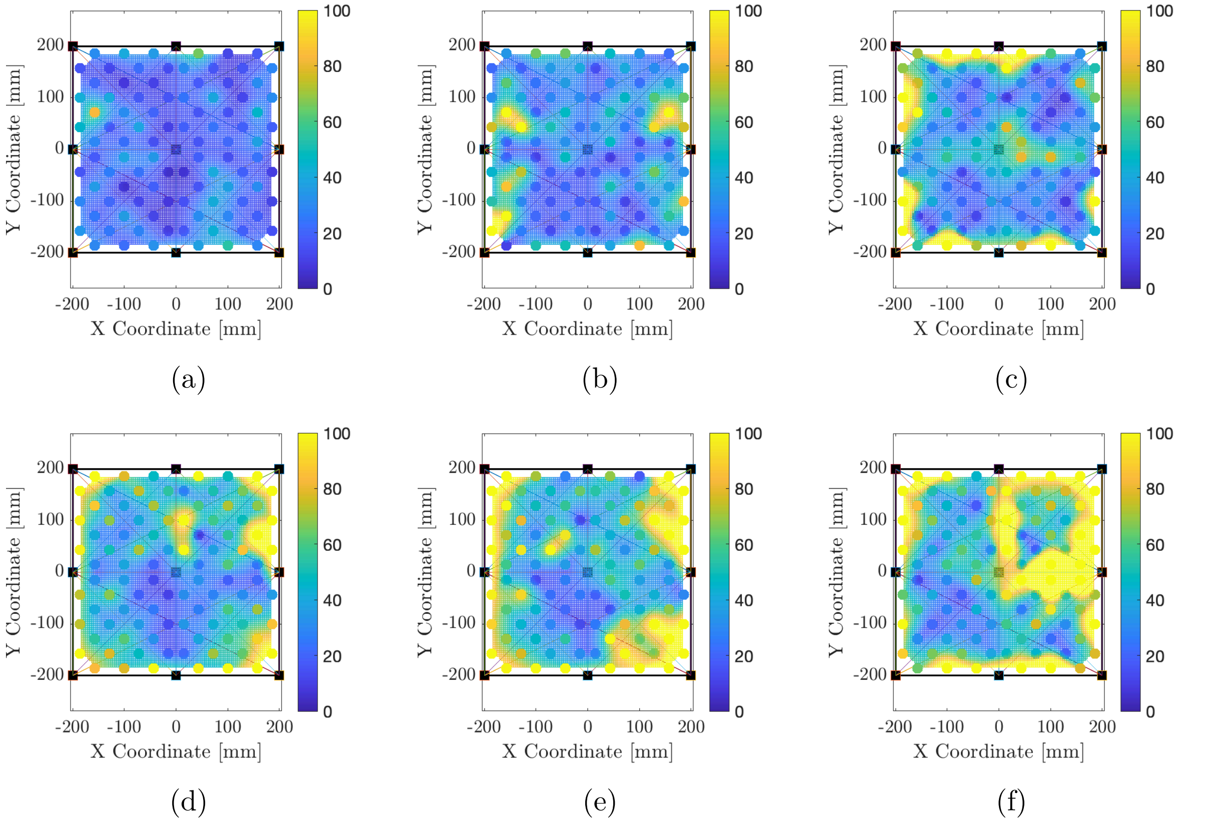
Figure 34 showcases the distribution of prediction errors for the networks as a fraction of the simulated damage diameter (1.25 inches). This metric is relevant because an error up to one damage diameter means that the predicted damage location would overlap with the actual damage on the structure, indicating a reliable prediction. In Fig. 34a, the majority of the 1D CNN's errors fall below one diameter for both validation and test datasets, with no prediction exceeding two diameters. In contrast, the MLP network exhibits a broader dispersion in its predictions, with the peak of its distribution is between one and two diameters, but the spread of errors extends up to six diameters, equating to an error of nearly 180 mm.



Source: Prepared by the author.

Figure 34 – Error distribution for validation and training datasets as a fraction of the diameter of the simulated damage used in the experiments: (a) 1D CNN; and (b) MLP network.

Figure 35 depicts the RMSE along the plate for the median network of both the 1D CNN and MLP architectures. In this representation, each point symbolizes a non-training damage position, with the colors signifying the magnitude of the prediction's error. These



(a) (b) (c)

(d) (e) (f)

Source: Prepared by the author.

Figure 35 – Spatial RMSE distribution for non-training data on the median network of both architectures and different amount of samples during training phase: (a) CNN using 25 training samples; (b) CNN using 16 training samples; (c) CNN using 9 training samples; (d) MLP using 25 training samples; (e) MLP using 16 training samples; and (f) MLP using 9 training samples.

figures highlight the impact of the data volume used during the training stage and provide insights into the nature of the errors. In Figs. 35a, 35b, and 35c, the 1D CNN network exhibits increased errors at the plate's borders and in regions crossed by a single signal path. Referring back to Fig. 24e, regions like the one near X > 150 mm and Y = 100 mm, which are only intersected by the direct line between PZTs 1 and 2, demonstrate significant errors. This pattern suggests that the 1D CNNs errors are predominantly due to insufficient information about the damage. Nevertheless, the 1D CNN trained with 9 samples per quadrant still demonstrates satisfactory performance when compared to the MLP network (refer to Figs. 35c and 35d).

In contrast, MLPs consistently exhibit poor performance across all scenarios. The MLP trained with 25 damaged positions per quadrant, for instance, underperforms compared to the 1D CNN trained with only 9 samples per quadrant, as evidenced by the high-error regions in the center of the plate in Fig. 35d. As the availability of training data decreases, the MLP's performance deteriorates significantly, with errors exceeding 100 millimeters in numerous areas as shown in Figs. 35e and 35f. This trend indicates the network's inability to effectively learn from the available data, likely due to the insufficiency of information in the DIs for accurate position mapping.

3.4 GENERAL GUIDELINES FOR NETWORKS ARCHITECTURES

One of the main challenges in the field of deep neural networks is determining the optimal network architecture and hyperparameters. Unlike traditional programming paradigms, there is no universally accepted methodology for identifying an ideal network configuration given an application. The selection of these parameters typically relies on the expertise of the programmer combined with a trial and error approach .

This Section introduces a series of guidelines and best practices for training neural networks, with a particular emphasis on 1D CNNs. These recommendations are based in the author's practical experience with these algorithms, supplemented by insights from current literature, and should be adapted for each application case. The motivation is to help the reader to minimize practical problems encountered while training neural networks with gradient descent methods. To facilitate comprehension, the guidelines are categorized into two main areas: network structure definition and training methodologies.

3.4.1 Network structure

- *Start with a simple architecture*

Initiate with a network of modest complexity, for instance, incorporating 3 to 4 convolutional blocks each followed by ReLU activation and a pooling layer, culminating in a single fully-connected layer. While it is tempting to use more intricate network architecture, such as multiple convolutional layers or advanced designs like residual networks, such complexities necessitate extensive data for effective training. This requirement often poses challenges in SHM contexts. Additionally, complex networks are more prone to issues like gradient vanishing and degradation. Consequently, in many instances, a network with simpler architecture can lead to superior results.

Regarding the convolutional layers, image processing networks predominantly utilize kernels sized 3×3 or 5×5. For Lamb wave processing, it is a good idea to evaluate the kernel size based on time signal discretization. Small kernels in a signal measured with a high sampling frequency may turn the focus of the network to a point by point analysis and require multiple convolutions+pooling operations to extract relevant features. Conversely, big kernels may encompass more than one wave period and could miss important local features on Lamb waves signals in damage scenarios, such as attenuation and phase shifts.

- *Avoid convolution with even size kernels*

Avoid using even kernel sizes for convolutional layers, e.g., 2, 4, etc.. Using even-sized kernels in convolutional layers can lead to a symmetry problem because they lack a central point, which is crucial for balanced feature extraction. In an even-sized kernel, like 2 or 4, the absence of this central point results in asymmetric processing of the input data. This asymmetry can skew the convolution operation to the right or to the left., as the kernel does not uniformly cover the area around a point. It causes alignment challenges and potentially biased feature detection, particularly in tasks where precise spatial/temporal relationships are important. Conversely, odd-sized kernels, such as 3 or 5, have a central point that ensures a symmetrical and consistent field of view, leading to more effective and balanced feature extraction in CNNs.

- *Avoid odd pooling kernels*

Employ pooling layers with thoughtfully selected kernel sizes and strides to minimize information loss. It is advisable to use pooling layers where the kernel size is a multiple of two, paired with an equivalent stride. For example, a max pooling layer with both kernel and stride set to 2 effectively reduces the data volume by 50%, resampling the resultant array by a factor of two. Similarly, a configuration with a kernel size and stride of 4 will discard 75% of the values.

Furthermore, utilize convolutional layers with ‘‘same’’ padding and pooling layers without padding. This configuration allows for dimension control through the pooling layer, with convolutional layers focusing on feature extraction and pooling layers on subsampling.

Additionally, adhering to two basic design principles makes it is easier to create a consistent layer size: (i) layers yielding feature maps (e.g. convolution layers) should maintain a consistent channel count; (ii) a reduction in feature map size should be followed by an increase in channel number, ensuring spatial/temporal complexity remains consistent across layers.

- *Adjust the input time series to be divided by two multiple times*

Building on the previous guideline, ensure the input time series length is divisible by powers of two (e.g., signals with 1024, 2048, 4096 points), achieved through truncating or resampling the Lamb wave signal. This setup enables the integration of multiple max pooling layers with varying kernel sizes such as 2, 4, 8, or 16. Coupled with the recommendation to avoid odd-sized pooling kernels, this strategy effectively reduces the number of parameters in the temporal dimension. Meanwhile, the convolutional layers incrementally adds complexity to the feature maps without increasing the parameter count in the final layer.

Consider the following two strategies, which apply different convolution and pooling approaches to the same input signal:

- **Strategy #1** maintains a constant channel count for convolution and kernel size for pooling layers.

Algorithm 4: Strategy #1 - Constant number of channels in convolution + constant pooling size

```
Input: Convolutional Layer with $F = 16$ channels, Pooling layer kerrnel size $H = 2$ and stride
       $S = 2$
8 signals with 1024 points → 8192 points;
Convolution with $F = 16$ → $8192 \times 16$ points;
Pooling with $H = 2$ and $S = 2$ → $4096 \times 16$ points;
Convolution $F = 16$ → $4096 \times 16$ points;
Pooling with $H = 2$ and $S = 2$ → $2048 \times 16$ points;
Convolution $F = 16$ → $2048 \times 16$ points;
Pooling with $H = 2$ and $S = 2$ → $1024 \times 16$ points;
Flatten layer → $1024 \times 16 = 16384$ points;
Fully-connected layer with $N = 50$ neurons → $50 \times 16384 = \mathbf{819200}$ parameters;
```

- **Strategy #2** begins with fewer channels and a larger pooling kernel, progressively increasing the convolutional channels and decreasing the pooling kernel size. This approach allows the initial layers to focus on finer signal details, which can then be aggregated or discarded through pooling (e.g., average pooling). As network depth increases, these features grow in number and complexity through successive convolutional layers, culminating in a few high-level features in the terminal layers.

Algorithm 5: Strategy #2 - increasing number of channels in convolution + decreasing pooling size

```
Input: Convolutional Layer with increasing number of channels $\boldsymbol{F}$, and decreading pooling
       layer kerrnel size $\boldsymbol{H}$ and stride $\boldsymbol{S}$
8 signals with 1024 points → 8192 points;
Convolution with $\boldsymbol{F} = \mathbf{4}$ → 8192 × 4 points;
Pooling with $\boldsymbol{H} = \mathbf{16}$ and $\boldsymbol{S} = \mathbf{16}$ → 512 × 4 points;
Convolution $\boldsymbol{F} = \mathbf{8}$ → 512 × 8 points;
Pooling with $\boldsymbol{H} = \mathbf{8}$ and $\boldsymbol{S} = \mathbf{8}$ → 64 × 8 points;
Convolution $\boldsymbol{F} = \mathbf{16}$ → 64 × 16 points;
Pooling with $\boldsymbol{H} = \mathbf{4}$ and $\boldsymbol{S} = \mathbf{4}$ → 16 × 16 points;
Flatten layer → 16 × 16 = 256 points;
Fully-connected layer with $\boldsymbol{N} = \mathbf{50}$ neurons → 50 × 256 = **12800** parameters;
```

Note that the parameter count presented primarily emphasizes the parameters of the final fully connected layer. This is because the number of parameters in the convolutional layers is small compared to the last layer. For example, a 1D convolutional layer with a kernel size of 3 and 16 channels has 48 weights (since each of the 16 channels has a 3x1 kernel, amounting to 3 weights per channel) and 16 biases, totaling 64 parameters. This difference in parameter distribution highlights the significant role of the fully connected layer in the overall parameter count of the network.

Although Strategies #1 and #2 are applied to the same signal and follow a similar architectural framework, they diverge significantly in terms of training dynamics. Strategy #1 culminates in a network that is approximately 64 times larger than that developed using Strategy #2. This substantial increase in network size increases the training time and elevates the risk of overfitting during the training phase. Furthermore, based on the author's experience, Strategy #1 tends to result in a network with inferior generalization capabilities compared to that derived from Strategy #2.

3.4.2 General training guidelines

Training neural networks using gradient descent algorithms involves multiple challenges, including optimal network initialization, learning rate adjustment, and regularization techniques. This section offers a set of training recommendations that reader should adjust to fit the specific requirements of their problem.

1. Network initialization: Commence with a basic neural network structure, adjusting key parameters such as learning rate and decay to understand computational load

and initial model responsiveness.

2. Learning rate optimization: Systematically modify the learning rate to analyze its impact on the model's loss function and convergence efficiency.

3. Overfitting assessment: Attempt to overfit the model to the training data, which serves as a benchmark for the network's capacity and initial effectiveness.

4. Learning rate decay: While trying to overfit training data, implement learning rate decay cautiously, ensuring it enhances model convergence without causing premature learning stagnation.

5. FC layer augmentation: In scenarios where loss minimization is insufficient, incrementally add neurons to the fully-connected layer to enhance model complexity and learning potential.

6. Layer expansion: If FC layer augmentation proves inadequate, integrate an additional block convolutional-ReLU-pooling, while closely monitoring loss trends for optimization. A well-configured network should demonstrate the ability to memorize training data.

7. Regularization techniques: Following the achievement of satisfactory training efficiency, shift focus to regularization strategies to improve model generalization.

8. Dropout layer integration: Incorporate a dropout layer subsequent to feature flattening to address overfitting and enhance model robustness. If the network keeps overtiffing, adds a dropout after the convolution to limit the model access to low level features.

9. Mini-Batch utilization: Employ small mini-batch sizes, facilitating implicit regularization and exposure to diverse data classes during training iterations.

10. Overfitting analysis: Employ histogram-based comparative analysis of model's results with respect to training and validation data distributions to quantitatively assess model overfitting.

11. Data Adequacy Evaluation: Persisting challenges may indicate limitations in input data quality or quantity. In cases of suspected input inadequacy, evaluate data quality and PZT arrays configurations to identify potential uncovered regions. In this phase, a map of validation error with respect to position (simmilar to Fig. 35) may be useful.

3.5 CONCLUDING REMARKS

The initial Section of this Chapter introduced the LaWaDe system, a platform developed by GRAVi-UFMG for acquiring Lamb wave signals. Hardware and software modifications implemented in this system enable automatic signal capturing, which significantly facilitates the collection of extensive datasets of Lamb wave signals. These datasets are essential for training machine learning algorithms.

Subsequently, a dataset comprising Lamb wave signals from damages in a carbon plate was detailed. This dataset, consisting of 196 simulated damage positions across four quadrants defined by 9 PZT sensors, allows for testing both global and local damage localization strategies, potentially applicable to larger structures. The preprocessing steps, including crosstalk removal, noise filtering, and signal windowing, were also discussed.

The chapter proceeded to use this dataset to assess a 1D CNN network for damage localization. Multiple network architectures were tested, and the best-performing, least complex model was chosen. The performance of the 1D CNN using Lamb wave signals was compared to a classical MLP network that utilizes DIs as input parameters. In all assessed tasks, the 1D CNN consistently outperformed the MLP network, demonstrating its effectiveness as a feature extractor, classifier, and regressor for Lamb wave signals. Additionally, both algorithms were evaluated across different data availability scenarios. While a decline in performance was observed for both algorithms with decreasing training data, the 1D CNNs notably maintained consistent performance levels. This characteristic was evident even in scenarios with significantly limited training data, for instance, when the sample size was reduced from 25 to only 9 samples per quadrant of the plate. On the down side, 1D CNNs are harder and more expensive to train when compared to classical MLP networks, i.e., they use more computational resources and the hyperparameter tunning step is harder. Finally, the chapter ends with guidelines and best practices for training neural networks based in the author's experience with these algorithms.

As demonstrated in this chapter, the effectiveness of neural networks heavily relies on the quantity and quality of the training data. Solely depending on experimental data collection for SHM applications is neither economically nor practically feasible. Thus, the availability of a reliable strategy for data generation is crucial for training these algorithms and a promising research field. The next chapter introduces an innovative approach for generating reliable training data for machine learning using FE models and Bayesian inference. This proposed framework incorporates experimental setup uncertainties into the FE model, enhancing its reliablility and applicability.

4 BAYESIAN UPDATING FOR A LAMB WAVE MODEL UNDER VARYING TEMPERATURE

The development of a 1D CNN-based SHM system, as described in Chapter 3, certainly showcased promising results, especially with its capability to accurately locate the damage position. However, there are always challenges and constraints to any experimental or computational work, and in this case, the time spent in data collection stands out as a primary concern.

Experimental data collection is a inherently complicated task. In the specific experimental application showcased in Chapter 3, the data collection involves damage positioning, capturing data, and saving it for each unique damage position on the 7x7 damage grid. When considering that a single position on this grid requires roughly 8 minutes for completion, the magnitude of the task becomes evident. It translates to approximately 8 hours of continuous work to cover the entire grid of one quadrant. Spread over a typical workday, the complete colection of damage on all four quadrants can extends to almost a working week just to account for a single damage size. Indeed, the inherent constraints of collecting data across multiple combinations of positions, damage sizes, types, temperature conditions, and excitation frequencies can be a significant limitation in creating a robust and versatile system, as the sheer amount of time and resources needed becomes practically unsustainable.

Hence, the need for an alternative, more efficient approach to data generation becomes important. One promising way, as evidenced by literature (refer to Section 2.2.3), is the use of numerically generated Lamb wave data to train machine learning algorithms. Such an approach not only reduces the dependency on labor-intensive experimental data collection but also offers flexibility in exploring a wide range of scenarios in a controlled computational environment. However, a critical aspect that seems underexplored in the literature is the uncertainty analysis associated with incorporating Lamb wave data into FE models. Model uncertainty quantification in computational simulations is crucial for model validity, robustness and confidence on predictions.

Therefore this chapter presents a framework for stochastic updating to verify and validate a FE model of a composite plate, considering the influence of temperature on Lamb wave propagation. It begins with a deterministic updating procedure to find optimal mechanical properties, followed by a stochastic updating procedure to obtain probability density functions for the meaningful parameters. The stochastic updating procedure is divided into two steps: a sensitivity analysis using Sobol Indices and a Bayesian inference process using Markov-chain Monte Carlo (MCMC) simulations and the Metropolis-Hastings sampling algorithm. To reduce the computational time required for the Monte Carlo process, this work proposes a surrogate model based on artificial neural networks. The proposed network can be trained using parallelized Monte Carlo simulations, in contrast to the sequential nature of the Markov-Chain process. The usage of the surrogate model reduced the time required for updating rounds by almost 500 times in the studied case without compromising the accuracy of the resulting probability density functions for model parameters.

The chapter is organized as follows: Section 4.1 provides a context for model updating techniques, presenting relevant literature that integrates Bayesian inference with Lamb waves simulation. Section 4.2 details the novel framework proposed, which aims to update finite element (FE) models by incorporating experimental uncertainty and the effects of temperature on simulations. Section 4.3 introduces the CONCEPT (CarbON-epoxy CompositE PlaTe) experimental dataset and describes the FE model developed to simulate it. Following this, Section 4.4 demonstrates the application of the updating framework to the FE model across various temperatures. This Section also outlines the development and training of a surrogate neural network model, designed to replace the FE model. Finally, Section 4.5 offers concluding remarks on the findings and their implications.

The work presented in this chapter is based on the article "Bayesian calibration for Lamb wave propagation on a composite plate using a machine learning surrogate model", published in the Mechanical Systems and Signal Processing journal, volume 208, in February 15, 2024.

4.1 CONTEXTUALIZATION

As discussed in Section 2.2.1, Lamb waves are ultrasonic-guided elastic waves widely used for SHM and assessing plate-like structures in various applications. Due to their ability to travel long distances, Lamb waves find applications in local inspections (fatigue (WANG *et al.*, 2018), corrosion (NAGY; SIMONETTI; INSTANES, 2014), bonded joints (TONG *et al.*, 2022)) and large-scale structures such as aerospace components (BOLLER; TOMLINSON; STASZEWSKI, 2004). This work focuses on Lamb wave propagation in composite structures (ZENG *et al.*, 2022), facing challenges from material heterogeneity and the influence of temperature and humidity on physical properties (KONSTANTINIDIS; DRINKWATER; WILCOX, 2006; SCALEA; SALAMONE, 2008).

Guided wave propagation simulation in composite structures is an active research area with various methods available, including semi-empirical and numerical techniques (WANG; YUAN, 2007a; MITRA; GOPALAKRISHNAN, 2016). Among numerical methods, the FE method stands out for its accurate representation of wavefront propagation in complex structures (LUCA *et al.*, 2020). However, high-frequency simulations or modeling small features require very fine mesh density, leading to computationally expensive models. To achieve a desirable degree of fidelity and to consider environmental variations, numerical models must be updated with experimental data. Ereiz, Duvnjak and Jimenez-Alonso (2022) present an extensive discussion about finite element model updating methods, that can be divided into deterministic and stochastic (or probabilistic) approaches.

Deterministic updating is usually encoded as a minimization problem and aims at finding point estimates, or optimal values, for physical parameters. These problems can be solved with techniques ranging from traditional optimization algorithms, like the least-squares method, to more advanced ones like genetic algorithms (STANDOLI *et al.*, 2021) or particle swarm optimization(MARWALA, 2010). Unfortunately, any model is an idealization of reality and, therefore, incapable of representing all features. FE models often have significant uncertainty in their responses because of the lack of knowledge about material and geometric properties as well as boundary and loading conditions, among other factors. Stochastic approaches take these uncertainties into account during calibration steps, and include them in the updating procedure.

Among stochastic approaches, one of the most used is the Bayesian updating framework (refer to Section 2.4.4). This approach uses Bayesian inference to infer information about a posterior probability distribution of model parameters based on a prior probability distribution, experimental data, and a likelihood function. To generate samples of the unknown posterior, several methods have been proposed, such as MCMC, Transitional Markov Chain Monte Carlo (TMCMC), and Sequential Monte Carlo sampler (LYE; CICIRELLO; PATELLI, 2021). Several works used Bayesian inference as a tool, including works with bolted joints (TELOLI *et al.*, 2021; MIGUEL; TELOLI; da Silva, 2022), vibration of beams with varying boundary conditions (RITTO; SAMPAIO; AGUIAR, 2016), and geometric non-linearities (WANG *et al.*, 2018). Due to the computational cost of the Bayesian inference process, multiple works used surrogate modeling techniques (WAN *et al.*, 2014; WAN; REN, 2016; LIU *et al.*, 2020; LIM; MANUEL, 2021; ZHOU *et al.*, 2016; FAKIH *et al.*, 2022; PADIL; BAKHARY; HAO, 2017).

In the context of a probabilistic perspective applied to composite structures subjected to Lamb wave excitation, Galina et al. (GALLINA *et al.*, 2017) proposed a methodology for mechanical properties identification based on Bayesian inference and semi-empirical dispersion curves obtained with a LISA model. Yan et al. (YAN *et al.*, 2020) used multiple frequency excitation to obtain experimental dispersion curves and developed a scheme based on WFE assisted metamodel. Then, combining a Kriging predictor with TMCMC rounds, they sampled posterior Probability Density Functions (PDF) for laminate properties.

Both studies share a common feature in the usage of specific Lamb wave simulation techniques such as LISA and WFE assisted method. These methods need to be manually implemented as they are not readily available on commercial analysis software. Also, neither of these works investigated the effect of temperature on the mechanical properties or the Bayesian inference process.

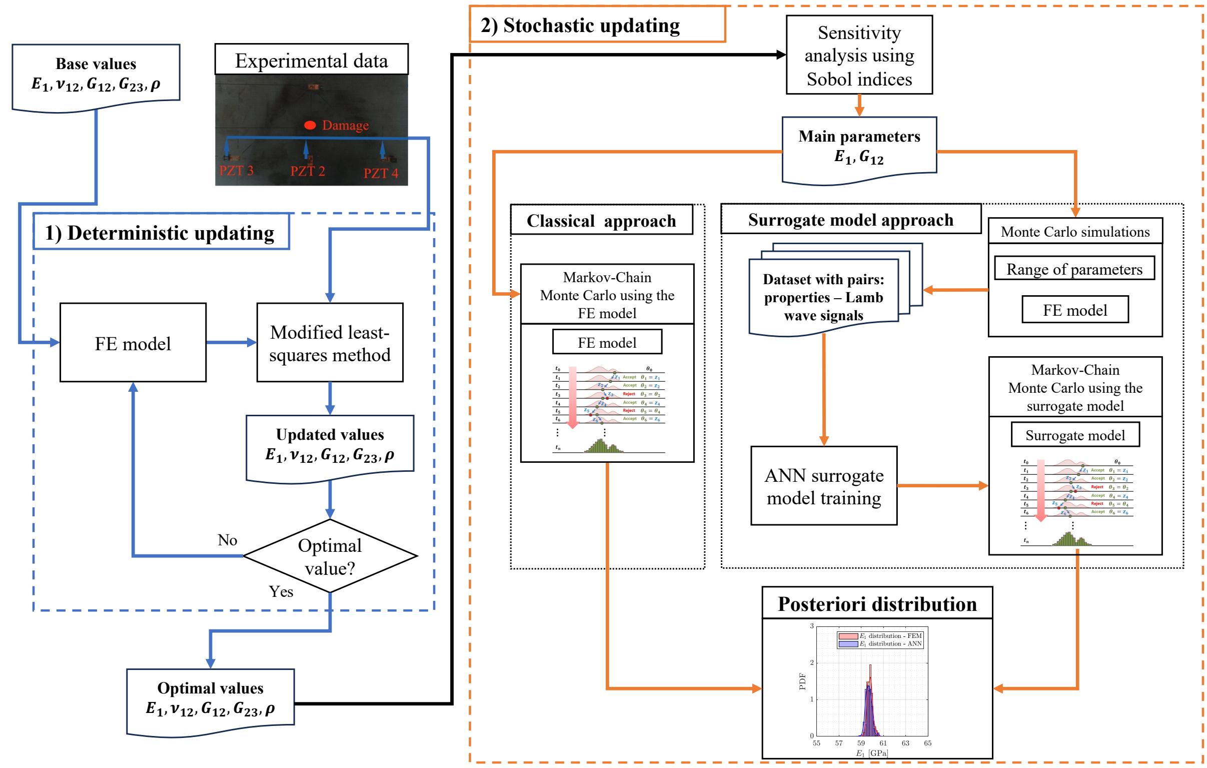
4.2 PROPOSED FRAMEWORK

As a contribution to the literature, this work proposes a framework for applying Bayesian inference to obtain a stochastic FE model for Lamb wave propagation under varying temperature. Additionally, the work also proposes the use of a machine learning surrogate model based on ANN to considerably reduce the time required for the MCMC rounds. The work also evaluates the impact of the surrogate model on the posterior distribution sampled by the MCMC, along with database size and simulation time for training the surrogate model.

The proposed framework for stochastic calibration of a FE model for Lamb wave propagation on a composite plate is performed for multiple temperatures, aiming to construct a model capable of emulating the whole experimental dataset. The framework lays its foundation on the following assumptions and considerations:

1. The database made available by da Silva et al. (SILVA *et al.*, 2020) is used as an experimental setup, and it is considered that no material property is known *a priori*.
2. All effects imposed by temperature on measured signals are incorporated at the material level.
3. A set of material properties is obtained among various possibilities using a deterministic updating procedure with a modified version of the least-squares method (BUD *et al.*, 2022). The deterministic version of model parameters provides the necessary information for the stochastic updating procedure.
4. A sensitivity analysis using Sobol indices to identify the key parameters that significantly impact the model results and should therefore be included in the Bayesian inference process.
5. A Bayesian inference process using MCMC simulations and the Metropolis-Hastings sampling algorithm is used to sample posterior probability functions from uniform prior distributions.
6. The classical FE model-based approach is replaced by a machine learning surrogate model approach based on neural networks to reduce simulation time.

The complete strategy is presented in Fig. 36, and the underlying theoretical foundations are discussed in the subsequent subsections.


2) Stochastic updating
Base values
$E_1, \nu_{12}, G_{12}, G_{23}, \rho$
Experimental data
Damage
PZT 3
PZT 2
PZT 4
1) Deterministic updating
FE model
Modified least-squares method
Updated values
$E_1, \nu_{12}, G_{12}, G_{23}, \rho$
No
Optimal value?
Yes
Optimal values
$E_1, \nu_{12}, G_{12}, G_{23}, \rho$
Sensitivity analysis using Sobol indices
Main parameters
E_1, G_{12}
Classical approach
Markov-Chain Monte Carlo using the FE model
FE model
Surrogate model approach
Dataset with pairs: properties – Lamb wave signals
ANN surrogate model training
Monte Carlo simulations
Range of parameters
FE model
Markov-Chain Monte Carlo using the surrogate model
Surrogate model
Posteriori distribution
E_1 distribution - FEM
E_1 distribution - ANN
PDF
0
1
2
3
55
57
59
61
63
65
E_1 [GPa]

Source: MCMC illustration adapted from Jin, Ju and Jung (2019).

Figure 36 – Proposed framework with deterministic and stochastic updating.

4.2.1 Deterministic updating procedure

This work uses the modified version of the least-squares method presented in Section 2.4.3 to perform an initial deterministic updating of input parameters. Three residuals based on the DIs proposed by Dworakowski *et al.* (2015) (refer to section 3.2.3) are established to measure the discrepancy between experimental and model signals. Note that although the equations presented are continuous, they are implemented using discrete numerical computations.

The first residual evaluates the squared norm of the difference between the experimental and numerical signals. This residual can quantify differences in amplitude and phase between the signals, and it is calculated according to the equation:

$$R_{\mathrm{NORM}} = \frac{\displaystyle\int_{t_1}^{t_2} [y(t) - x(t)]^2 dt}{\displaystyle\int_{t_1}^{t_2} x(t)^2 dt}, \tag{34}$$

where $x(t)$ is the experimental signal and $y(t)$ is the numerical signal.

The second residue is based on the cross-correlation between the signals, and it is calculated as:

$$R_{\text{XCOR}} = 1 - r_{xy}, \tag{35}$$

where r_{xy} is the cross-correlation between the input signals and the signal at zero delay. This measure is only sensitive to changes in the shape and frequency of the signal. The third residual is the difference between the power spectra of the signals. This index is particularly useful for detecting delays and frequency shifts in the signals, and it is calculated as follows:

$$R_{\text{PSD}} = \frac{\int_{f_1}^{f_2} [Y(f) - X(f)]^2 df}{\int_{f_1}^{f_2} X(f)^2 df}, \tag{36}$$

where $X(f)$ is the power spectral density of the experimental signal and $Y(f)$ is the the power spectral density of the numerical signal.

The final residual is defined as the weighted sum of the individual residuals:

$$R_{\text{TOTAL}} = k_1\ R_{\text{NORM}} + k_2\ R_{\text{XCOR}} + k_3\ R_{\text{PSD}}. \tag{37}$$

where k_1, k_2, and k_3 are adjustable weights assigned to each residual. These weights allow for controlling the relative importance of each metric in the outcome of the least-squares method.

4.2.2 Stochastic updating procedure

Sensitivity analysis

Sensitivity analysis aims to uncover the effects of model parameters, which are referred to as random input variables, on the quantities of interest that constitute the model response. To assess the influence of these parameters, first and second-order Sobol indices (as discussed in Section 2.4.1) are utilized. These indices allow evaluation of the global impact of the parameters, and their values are obtained through Monte Carlo simulations. Specifically, the UQLab framework is employed for this purpose (MARELLI; SUDRET, 2014).

Bayesian inference

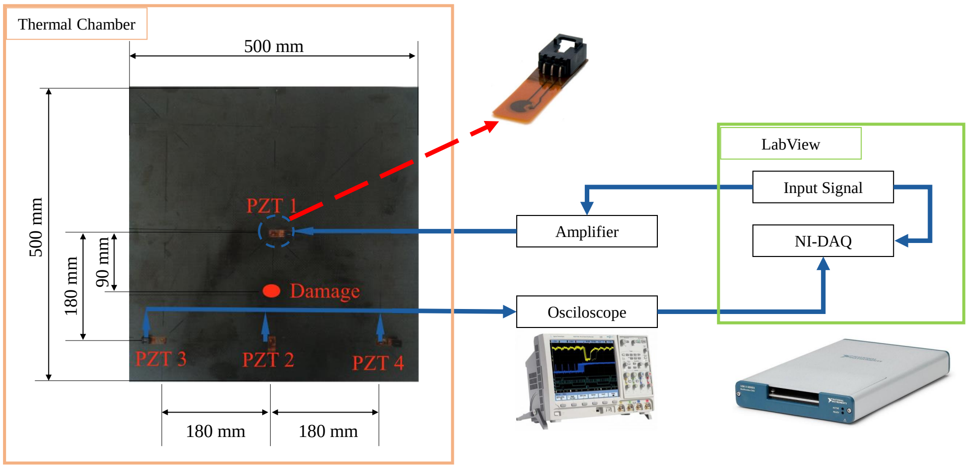
Physical systems exibit inherent variations, e.g., geometry, environmental conditions, material properties, etc. Thus, even the most complete data set cannot eliminate this uncertainty, so-called irreducible. To include this type of variability in the parameters of the computational model, this work adopts the Bayesian paradigm as a statistical inference tool for the model calibration problem (refer to Section 2.4.4). Two assumptions are introduced: (i) As the model parameters are random variables, a prior distribution $\pi(\boldsymbol{\theta})$ for the set of input variables $\boldsymbol{\theta}$ is proposed based on the deterministic model updating procedure; (ii) Bayes' theorem is employed to update the prior to a posterior distribution, gathering information from random data observations $(\mathscr{D})$. The posterior density function (PDF) is sampled using the MCMC/Metropolis-Hastings algorithm.

The random variables $\boldsymbol{\theta}$ are limited to the interval $[\boldsymbol{\theta}_{\text{min}}, \boldsymbol{\theta}_{\text{max}}]$, whose current state is symmetrically normalized as $\boldsymbol{\theta'} = (1 - \boldsymbol{x})\boldsymbol{\theta}_{\text{min}} + \boldsymbol{x}\boldsymbol{\theta}_{\text{max}}$; $\boldsymbol{x}$ is a random variable $\in [0, 1]$ that represents generated candidates for the posterior distribution. These candidates are sampled from a normal distribution with standard deviation σ_p. This hyperparameter controls the random walk step to avoid the chain becoming static and, at the same time, to allow the parameter space to be properly explored (LYE; CICIRELLO; PATELLI, 2021). Only 80% of the Monte Carlo simulations are considered in the final stationary Markov chain (burn-in of 20%).

4.3 CASE STUDY

4.3.1 CONCEPT: CarbON-epoxy CompositE PlaTe

The experimental setup shown in Fig. 37 consists of a 500 x 500 x 2 mm^3 carbon fiber reinforced polymer (CFRP) plate with 10 layers of plain weave fibers. The plain weave fibers are oriented along the plate borders' directions. Four PbZrTi (Lead Zirconate Titanate - PZT) SMART Layers from Accelent Technologies, each with a diameter of 6.35 mm, are bonded to the plate using epoxy resin. PZT 1 is the actuator, while PZT 2, PZT 3, and PZT 4 are sensors. Due to the dispersive nature of Lamb waves, the experiments are performed with a uniform excitation frequency across all tests, targeting a specific segment of the Lamb wave frequency-thickness spectrum. When considering the multimodal characteristic of Lamb waves, Wang and Yuan (2007b) suggest that for a plate of this nature, excitation of higher-order modes (A1, S1, A2, etc.) is expected to occur above 500 kHz. Therefore, the excitation signal is a 5-cycle sinusoidal wave



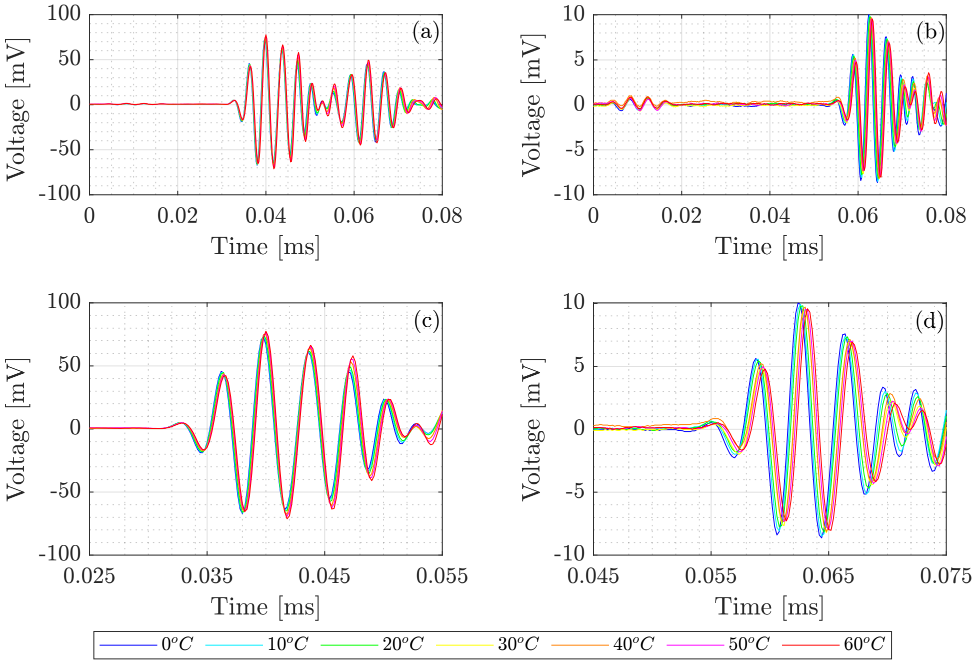
Source: Plate image adapted from Silva *et al.* (2020).

Figure 37 – Experimental Setup.

modulated by a Hanning window, with an amplitude of 35V and a central frequency of 250 kHz, aiming to excite only S0 and A0 modes. The output is recorded with a sampling frequency of 5 MHz over a 100 μs window using a NI USB 63533 from National Instruments and an oscilloscope DSO7034B from Keysight. A comprehensive exploration of this experimental setup is presented by França (FRANCA, 2014).

Data acquisition is performed at controlled varying temperatures ranging from 0°C to 60°C in increments of 10°C with a thermal chamber from Thermotron. For each temperature, 100 tests are repeated for statistical characterization. Silva *et al.* (2020) presents the complete description of the experimental procedure. Figure 38 shows one experimental measurement for each temperature from 0°C to 60°C. The experiments are also performed in progressive damage scenarios, with mass added to the region shown in red in Fig. 37. However, the present study only considered the plate in its undamaged condition. The dataset is publicly available in the GitHub[1] repository CONCEPT: CarbON-epoxy CompositE PlaTe.

[1] https://github.com/shm-unesp/DATASET_PLATEUN01



Source: Prepared by the author.

Figure 38 – Experimental results for temperatures from 0°C to 60°C. (a) Complete signal PZT 2; (b) complete signal PZT 3; (c) zoom at first signal package PZT 2; and (d) zoom at first signal package PZT 3.

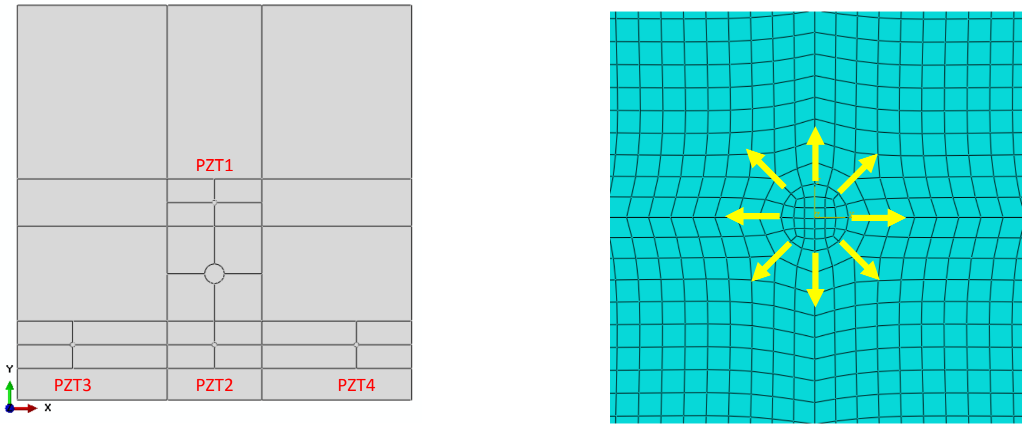
4.3.2 FE model

A FE model is implemented using ABAQUS/ Explicit. The plate is modeled using continuum shell elements (SCR8). These elements are designed to discretize a three-dimensional body rather than just representing a surface like traditional shell elements. Continuum shells have only displacement degrees of freedom, use linear interpolation, and include effects of transverse shear deformation and thickness change. Additionally, they are based on first-order composite theory and can be stacked to improve the accuracy of the through-thickness response.

The composite material is implemented in the FE model using ply-based properties. The layup is modeled as an orthotropic laminate property with three integration points per lamina. The discussion on the base properties and material considerations can be found in Section 4.4.1.

The spatial and temporal resolution of the model greatly affects the stability of the simulation. According to Moser (MOSER; JACOBS; QU, 1999), a spatial resolution of at least 20 nodes per wavelength is necessary to ensure stability. This can be written as $l_e = \lambda_{min}/20$, where λ_{min} is the minimum wavelength expected at the model and l_e is the characteristic length of the mesh. Additionally, to prevent numerical instability, the time step must be based on the maximum expected frequency in the simulation, with a recommended value of $\Delta t = 1/(20 f_{max})$, where Δt is the time step and f_{max} is the maximum frequency expected at the model. For an input frequency of 250 kHz, both these considerations result in a mesh with an element size of approximately 2 mm and a maximum time increment of 0.2 μs. However, further stability investigations of the model have shown that a time increment of 0.1 μs is needed to account for mesh irregularity around the PZT regions, ensure convergence, and keep the simulation time reasonable. The simulation window is set to 0.1 ms as this is the time necessary to capture the first two wave packets from Fig. 38.

Figure 39 presents the FE model. To facilitate the generation of a structured mesh for the entire plate, the plate is divided into several smaller rectangular regions, as shown in Fig. 39a. The area where the PZT actuators and sensors are attached to the plate is specifically modeled with a circular mesh. As proposed by Gresil et al. (GRESIL *et al.*, 2013), the excitation signal is applied as eight self-balancing forces around this region, as depicted in Fig. 39b. To obtain the output signal, the integrated strain results within the sensor region are used and transformed to voltage as suggested by Sirohi and Chopra (SIROHI; CHOPRA, 2000). The region between PZTs 1 and 2 is modeled with a circular mesh to account for damage addition in future model versions. However, in the present chapter, this region has the same element, materials, and conditions as the rest of the plate. Chapter 5 works with the damage model.



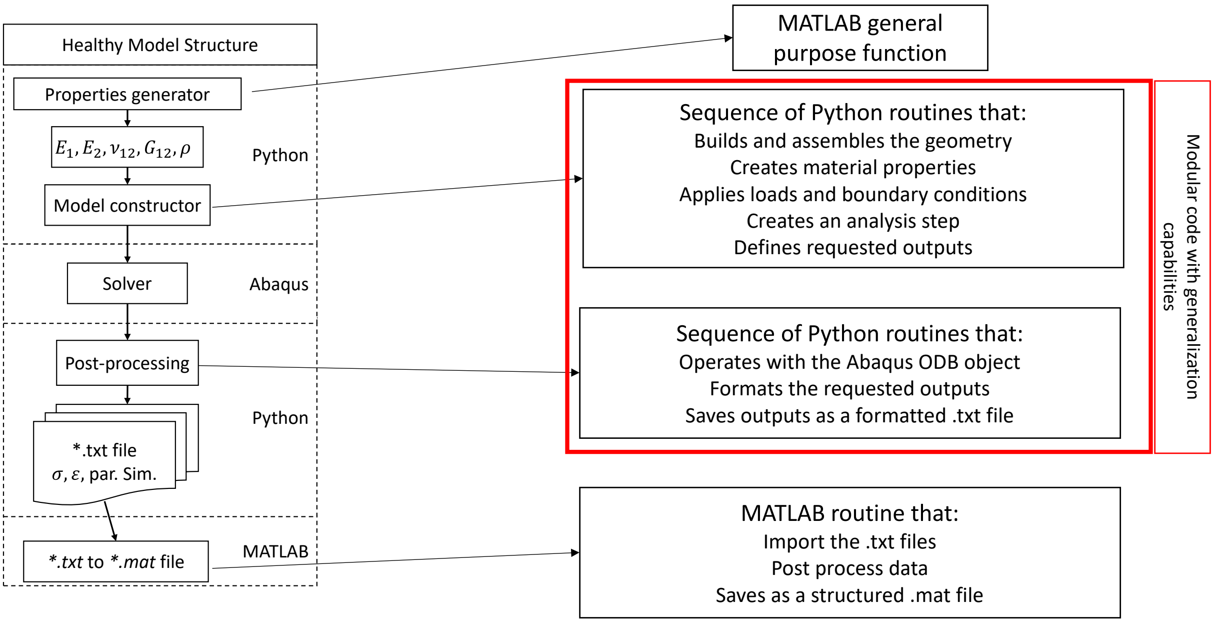
(a)

(b)

Source: Prepared by the author.

Figure 39 – Finite element model. (a) Model partition for mesh generation; and (b) PZT-1 showing excitation signal

The FE model is implemented using a series of Python routines that run within the Abaqus software, allowing for parametric simulation of the system. A code layer is also implemented using MATLAB and Windows command prompt, enabling direct control over Abaqus through the command line. This architecture made possible a complete automation of the pre-and post-processing steps of the simulation inside MATLAB's workspace. It also facilitated the usage of multiple machines to parallelize the Monte Carlo simulations. Figure 40 presents a description of how the routines are implemented.


Healthy Model Structure
Properties generator
$E_1, E_2, \nu_{12}, G_{12}, \rho$
Python
Model constructor
Solver
Abaqus
Post-processing
Python
*.txt file
σ, ε, par. Sim.
*.txt to *.mat file
MATLAB
MATLAB general purpose function
Sequence of Python routines that:
Builds and assembles the geometry
Creates material properties
Applies loads and boundary conditions
Creates an analysis step
Defines requested outputs
Sequence of Python routines that:
Operates with the Abaqus ODB object
Formats the requested outputs
Saves outputs as a formatted .txt file
Modular code with generalization capabilities
MATLAB routine that:
Import the .txt files
Post process data
Saves as a structured .mat file

Source: Prepared by the author.

Figure 40 – Python and MATLAB algorithm structures used to build, solve and post-process the FE model.

4.4 STOCHASTIC MODEL UPDATING FRAMEWORK: APPLICATION

4.4.1 Deterministic updating

The goal of the deterministic updating process is to bring the response signal of the FE model ($y(t)$) for sensors 2 and 3 as close as possible to the experimental results ($y_{\mathrm{EXP}}(t)$), by adjusting the mechanical properties of the materials. The FE model is symmetric concerning the horizontal and vertical axes, resulting in identical signals from PZTs 3 and 4. For this reason, only the PZT-3 signal is considered along with PZT-2 in the adjustment of mechanical properties. As the composite plate used in this work is obtained from a research project in partnership with a Brazilian aerospace company, which could not disclose the specific material or fabrication process used due to intellectual property agreements, the only information provided is that the plate is made of a prepreg aerospace-grade carbon fiber with 10 layers in a plain weave configuration (FRANCA, 2014). Thus, the initial range of values for the mechanical properties had to be determined by literature review, as detailed below.

The mechanical behavior of the materials is modeled using the classical laminate theory, where each layer is considered an orthotropic material. As such, Young's modulus in two perpendicular directions in the plane (E_1 and E_2), Poisson's ratio (ν_{12}), shear moduli in and out of a plane (G_{12} and G_{23}, respectively), and density (ρ) had to be determined. It is necessary to establish an initial search range for these parameters. This range is determined based on information from the literature for plain weave carbon/epoxy systems, with the main sources being the Composite Materials Handbook (CMH) (COMPOSITE..., 2017) and the work of Paiva et al. (PAIVA; MAYER; REZENDE, 2006). Table 5 presents the limits, initial and optimal values used for the following mechanical parameters: E_1, E_2, ν_{12}, G_{12} and ρ.

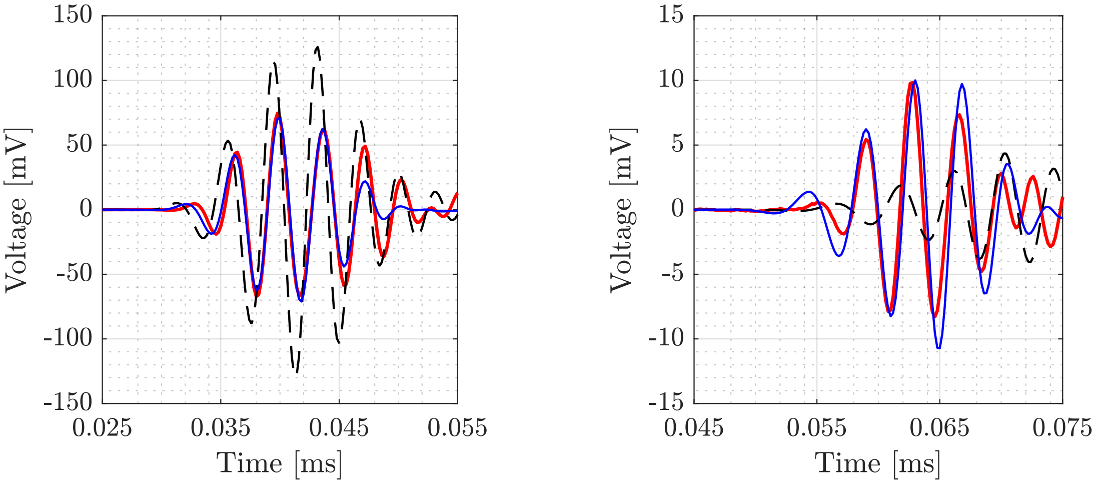
Table 5 – Parameter limits and initial values utilized for the deterministic updating procedure, including resultant optimal values.

Parameter	**Min**	**Max**	**Initial Value**	**Optimal Value**
E_1 [GPa]	60.0	80.0	70.0	61.2
E_2 [GPa]	60.0	80.0	70.0	61.2
ν_{12}	0.05	0.15	0.10	0.075
ρ $[Kg/m^3]$	1550	2250	1800	1620
G_{12} [GPa]	5.00	15.0	7.50	10.25
G_{23} [GPa]	4.00	6.00	5.00	5.00

Source: Prepared by the author.

As a simplification, it is assumed that the values of E_1 and E_2 for plain weave fibers are equal. The values presented in Tab. 5 are used as reference and adjusted using the iterative process based on least-squares method described in Bud et al. (BUD *et al.*, 2022). In each iteration, the model is initially simulated with the value of the properties from the former iteration. The residue between the experimental and model signals is calculated using Eq. (37). Next, each property is individually perturbed to determine the partial derivatives of the model response relative to inputs. The perturbation value is initially set to 5% of the current value and reduced over the optimization process to minimize system instability as the optimal values are approached. This process is repeated until the value of the residue converges. Only the first wave package in each channel is considered for the fitting. In the case of PZT 2, the second package shown in Fig. 38 represent edge reflections.

The oscillatory nature of the wave signal results in the residuals having multiple local extrema throughout the simulation range. For instance, if the simulated wave is one wavelength in advance or retard concerning the experimental data, the residual gives a local minimum. To mitigate this, the weights k_1, k_2, and k_3 in Eq. (37) are adjusted. Empirical testing has shown that weighting these parameters as 0.2, 0.4, and 0.3 can reduce the likelihood of becoming trapped in local minima, resulting in a better output signal. Also, intermediate solutions that give physically unrealistic properties are disregarded. Table 5 presents the initial and estimated values for the physical parameters of the material, whereas Fig. 41 compares the model results before and after the adjustment process. It is possible to observe that the adjustment process improved both the phase and the amplitude of the signals from PZTs 2 and 3.



(a) (b)

Source: Prepared by the author.

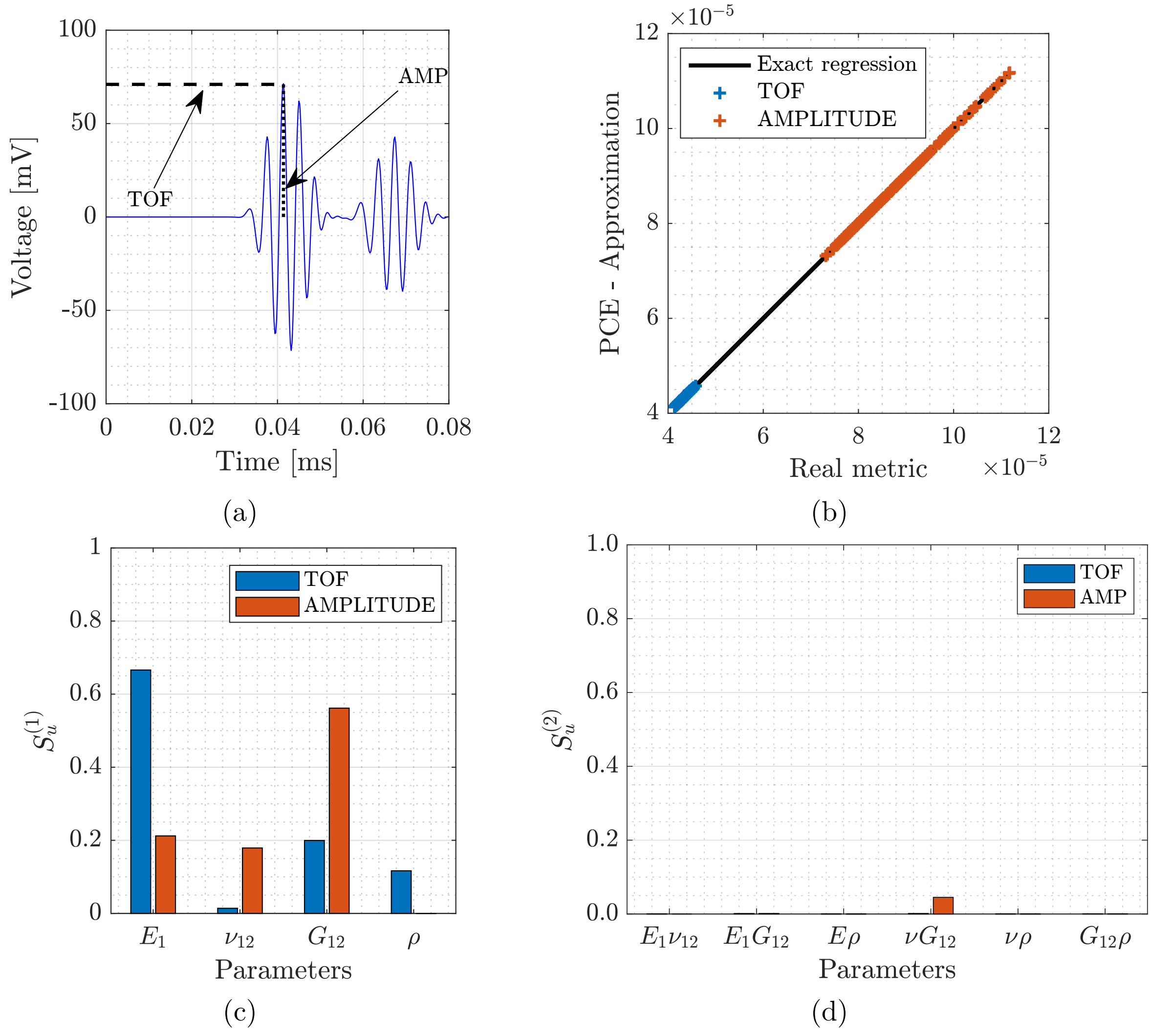
Figure 41 – Comparison between experimental (—) and numerical results with initial (—) and optimal (—) parameters. (a) Signal from PZT 2; and (b) signal from PZT 3.

Note that all parameters are well accommodated within the proposed ranges. For out-of-plane, the shear modulus G_{23} maintained its initial value even after optimization. Its variation within the established range does not influence the model performance - therefore, this parameter is kept fixed at 5 GPa in subsequent steps.

4.4.2 Stochastic updating

Sobol analysis

Following the deterministic adjustment, a sensitivity analysis is performed using Sobol indices to quantify each parameter's influence on the model's behavior. The parameters E_1, ν_{12}, G_{12}, and ρ are perturbed around the optimal deterministic values presented in Tab. 5. This perturbation is performed with 300 Monte Carlo simulations, in which model input parameters are sampled from a uniform distribution centered at the optimal values and bounded by ±20% limits. Two scores are defined to quantify the effect of the input parameters on the response of the model: the time of flight (TOF) and the maximum value of the envelope (AMP) of the first wave packet, as shown in Fig. 42a. Using the input parameters and these metrics, a PCE based metamodel of order 4 for the TOF and order 6 for the AMP metrics is implemented using the UQLab framework. Fig. 42b presents the validation of the PCE models using 20% of the samples that are not used for defining the model. Note that the PCE approximations for the metrics alingn with the exact regression for the model application range.



(a)

(b)

(c)

(d)

Source: Prepared by the author.

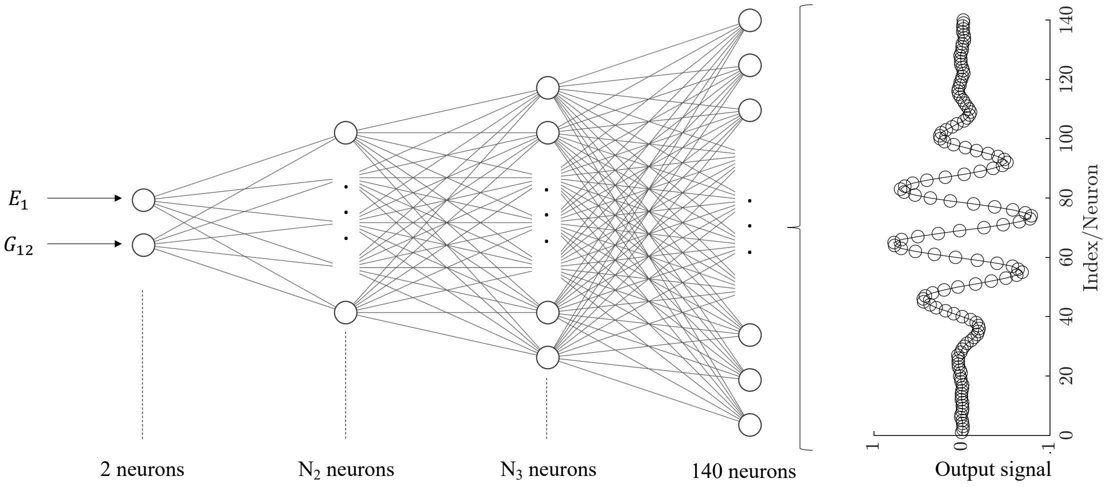
Figure 42 – Sobol indices analysis using PCE metamodel: (a) TOF and AMP metrics; (b) PCE metamodel validation; (c) first-order Sobol indices; and (d) second-order Sobol indices

The first and second orders of Sobol indices are shown in Figs. 42c and 42d, respectively. Note that the parameters with the greatest impact on the results are E_1 and G_{12}, mainly affecting approximately 70% and 55% of the variance related to TOF and amplitude, respectively. The sensitivity analysis indicates that ν and ρ have smaller influences when compared to E_1 and G_{12}. The combinations of parameters also have a negligible effect, with νG_{12} representing approximately 5% of the total variance related to the amplitude. Therefore, ν and ρ can be considered determined quantities, and their values are assigned based on the optimization results from Tab. 5, i.e., $\nu = 0.075$ and $\rho = 1620$ kg/m^3. The remaining parameters are considered undetermined, and updating their posterior probability distributions and gathering information from experimental data containing environmental variation is necessary. In summary, small intrinsic variations associated with uncertainties in the experimental data, which cause a slight change in the response pattern for the same test conditions, can influence the values identified for these parameters. Therefore, it demonstrates that in the presence of uncertainties in the experimental data, deterministic estimation of these parameters cannot be generalized - they must be treated as random variables.

4.4.3 ANN surrogate model

To accelerate time-consuming MCMC simulations, a surrogate model can replace the FE-based model during the Bayesian inference process. Some surrogate modeling techniques available in the literature include Gaussian processes (WAN *et al.*, 2014; WAN; REN, 2016), polynomial chaos expansion (LIU *et al.*, 2020; LIM; MANUEL, 2021), radial basis functions (ZHOU *et al.*, 2016), neural networks (FAKIH *et al.*, 2022; PADIL; BAKHARY; HAO, 2017), among others. This study employs a multilayer perceptron ANN as the surrogate model, serving as an interpolation model within the parameter search space for the Bayesian inference process. The input layer consists of two neurons, representing the two random variables E_1 and G_{12}, which are used as inputs to the surrogate model. The output layer comprises 140 neurons, representing the time series output. The number of hidden layers, as depicted in Fig. 43, is set to two, with N_2 and N_3 neurons in the first and second hidden layers, respectively.

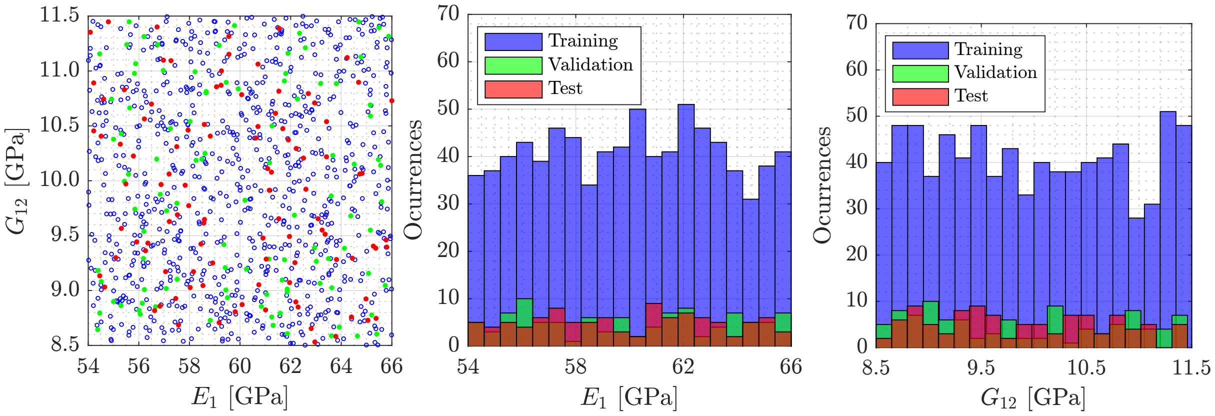
To train the surrogate model, the FE model is extensively simulated with properties sampled from bounded uniform distributions $E_1 \sim \mathcal{U}\big(54, 66\big)$ [MPa] and $G_{12} \sim \mathcal{U}\big(8.5, 11.5\big)$ [MPa] to generate 1024 samples. These bounds encompass the deterministic optimal values and offer sufficient margin to accommodate variations in the physical properties with temperature. During training rounds, data are divided with a proportion of 80/10/10, i.e., 820 samples for training, 102 for validation, and 102 for testing. The input parameters are normalized between 0 and 1, and the output signal is normalized between -1 and 1 to improve convergence. The model signal is truncated after the first wave packet to reduce



Source: Prepared by the author.

Figure 43 – Surogate model architecture

training time and the amount of data needed for network convergence. This packet has a group speed of 5920 m/s, within the range expected for the first symmetric mode of a carbon/epoxy composite and for the tested frequency/thickness ratio (WANG; YUAN, 2007b), and the second packet consists of its border reflection. Figure 44a-c shows the distribution of the sampling space population.



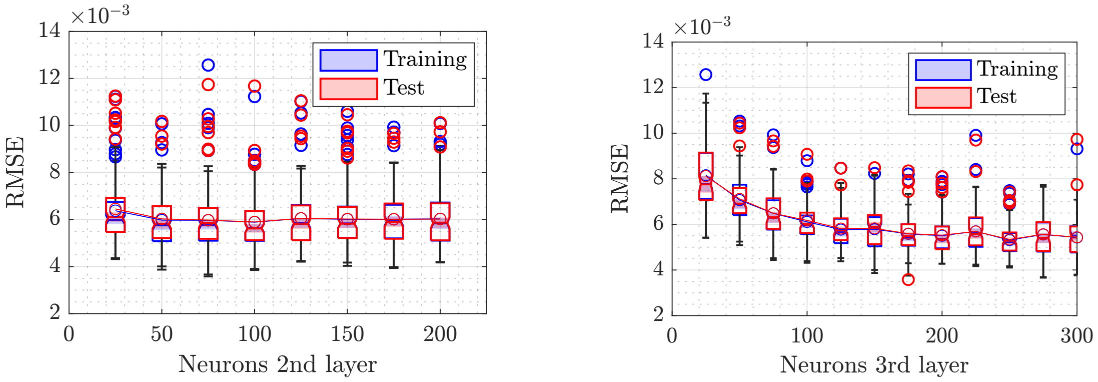
(a) (b) (c)

Source: Prepared by the author.

Figure 44 – Training (🟦), validation (🟩) and test (🟥) data distribution: (a) 2D dispersion; (b) E_1 Histogram; and (c) G_{12} Histogram

The surrogate model is trained using the Stochastic Gradient Descent with Momentum (SGDM) algorithm, and the network loss on validation data is employed as the stopping criterion for training. The discrepancy between the output of the surrogate model and the full-order FE model is the error metric guiding the learning process. At regular intervals, the training algorithm assesses the validation data, and if the loss on the validation data ceases to decrease or begins to increase, the training process is halted.

The performance of the proposed surrogate model is influenced by its hyperparameters, such as the number of neurons in the hidden layers, activation functions, initial learning rate, and the amount of available training data. N_2 and N_3 are defined through an exploratory search within 50 to 300 neurons per layer. Figure 45 presents the root mean square error (RMSE) concerning the number of neurons in the second and third layers. The RMSE is not significantly influenced by N_2, remaining stable with 50 or more neurons. In contrast, the number of neurons N_3 has a greater influence on the RMSE, with the error stabilizing with 200 or more neurons. Therefore, an architecture with 50 neurons for N_2 and 200 neurons for N_3 is defined.



(a)

(b)

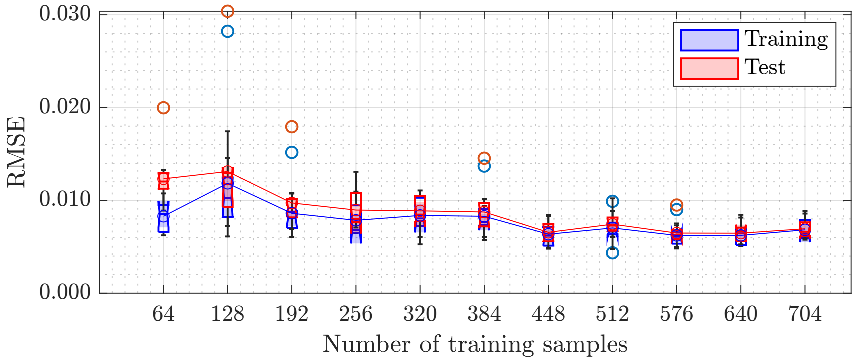
Source: Prepared by the author.

Figure 45 – Training (■) and test (■) RSME is a function of the number of neurons on: (a) the second layer; and (b) the third layer

The training process is repeated with varying sample sizes to evaluate the network's convergence regarding the training data. Figure 46 demonstrates that the RMSE decreases on both the training and test data as the number of training samples increases and eventually converges. Additionally, note that approximately 500 samples are the minimum required for the surrogate model to converge, i.e., less than the 820 samples used.

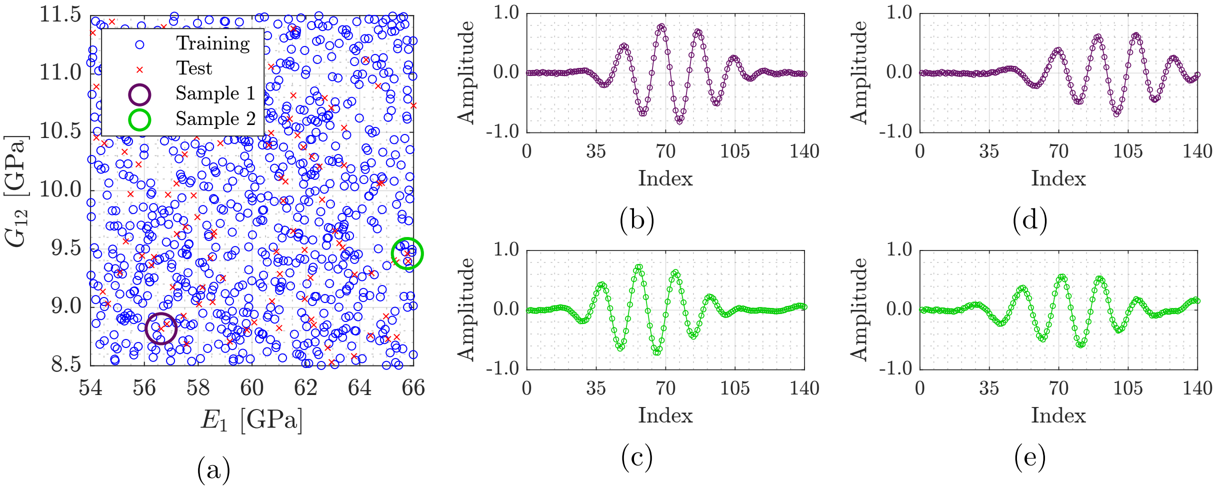
Figure 47 presents the results from the surrogate model compared to the FE model in three values of E_1 and G_{12} randomly selected from the test data. In these three samples, the surrogate model's response replicated the FE model's response without noticeable differences. Additional tests showed that this behavior is repeated throughout the entire training range.

In summary, the main advantages of the surrogate model are (i) its computation speed and (ii) the possibility of parallelizing the data-gathering step across multiple computers. For instance, each simulation using the FE model takes two to three minutes on a typical



Source: Prepared by the author.

Figure 46 – Training (■) and test (■) RSME with varying amounts of training samples



Source: Prepared by the author.

Figure 47 – Results for surrogate model performance on test data compared to FE model. (a) Subspace of surrogate model training and test data with two randomly selected test samples highlighted for model evaluation; (b) and (c) represents PZT 2 signals for samples 1 and 2, respectively; (d) and (e) represents PZT 3 signals for samples 1 and 2, respectively. On signal graphs (b) to (e), full lines (—) represent FE model predictions and circles (∘) represent ANN model results for the same sampled properties

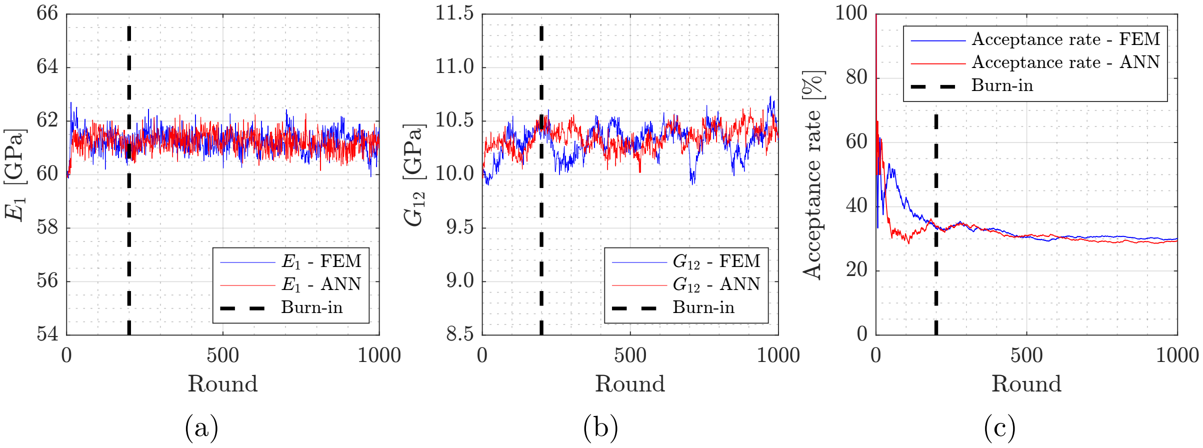
PC with an Intel core i7 CPU (3,0 MHz and 4 cores) and 16 GB of RAM, whereas the surrogate model takes less than a second to evaluate. The 1024 simulations used at the training step took around 30 hours to perform (without a parallelization scheme). After training, the surrogate model can be used for interpolation purposes within the domain defined for E_1 and G_{12}. In the current application, the same surrogate model can be used for all MCMC rounds at the temperature interval from 0°C to 60°C, reducing exponentially the amount of simulations and time required for the Bayesian inference process.

4.4.4 Bayesian inference

E_1 and G_{12} updating for $20°C$

The framework for the stochastic update process proposed in Section 2.4.4 is first applied to the experimental data collected at 20°C. The update process employs FE and surrogate models under this specific temperature condition. The outcomes of the two approaches are then compared, considering both the random walk performance and the updated posterior probability density function (PDF). This comparison employs the surrogate model to evaluate the uncertainty introduced in the Bayesian inference process.

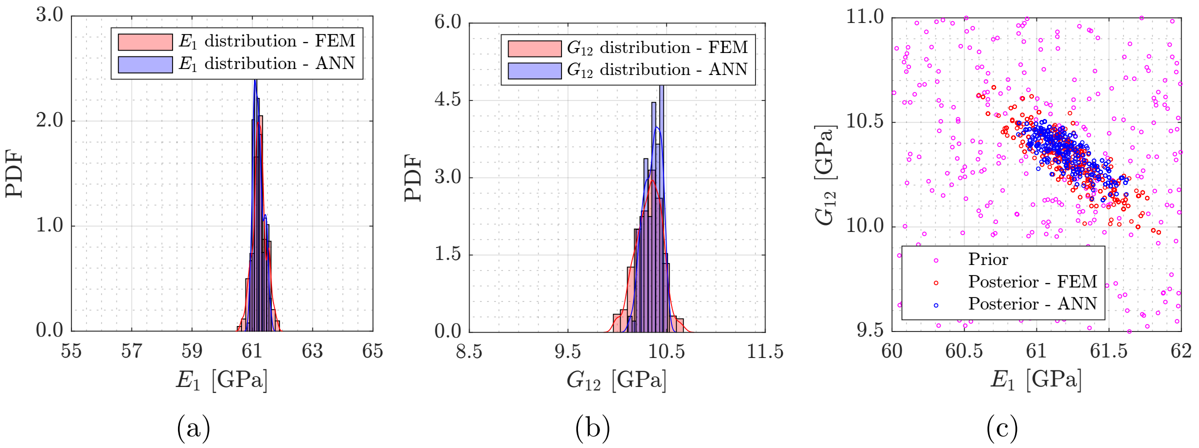
The prior probability distributions for the variables $\Theta = \{E_1, G_{12}\}^T$ are defined as $E_1 \sim \mathcal{U}(54, 66)$ [MPa] and $G_{12} \sim \mathcal{U}(8.5, 11.5)$ [MPa] to encompass the optimal values from the deterministic fitting (refer to Section 4.4.1) and provide a search range for the random walk algorithm. The initial value for each parameter is defined as the center of the search range $\Theta_i = \{60, 10\}^T$. In the Markov Chain estimation, the likelihood function's variance is iteratively defined through the fitting algorithm's rounds, aiming to ensure a stable acceptance rate between 15 and 50% (LYE; CICIRELLO; PATELLI, 2021). As illustrated in Fig. 48, the behavior of the FE and surrogate models is similar throughout the evolution of the Markov chain.



Source: Prepared by the author.

Figure 48 – Comparison between the random-walk performance of the MCMC algorithm while searching for E_1 and G_{12} using the numerical and surrogate models at 20°C. (a) Random walk values for E_1; (b) Random walk values for G_{12}; and (c) Acceptance rate for each strategy.

The random walk starts at the center of the search space for each parameter, and the Metropolis-Hastings sampling algorithm generates a new set of samples iteratively by proposing slight changes to the current set. Following the Metropolis-Hastings acceptance criteria, the proposed samples are accepted or rejected based on their likelihood compared to the current values. After several iterations, and upon proper adjustment of the MCMC parameters such as variance and step size for each round, the algorithm converges to an optimal region. This convergence is clearly illustrated in Figs. 48a and 48b. Within this stability region, the algorithm persists in sampling from the posterior distribution. In instances where the algorithm deviates significantly from the optimal region for the parameters, the proposed values become less likely to belong to the posterior distribution. As a result, the Metropolis-Hastings sampling algorithm tends to return to the region where it is more likely to acquire valuable information. The initial portion of the samples, which may exhibit transient behavior, can be discarded through a process known as "burn-in." This study sets the burn-in at 20% based on the acceptance rate curve presented in Fig. 48c. Figure 49 compares the posterior PDFs sampled for E_1 and G_{12} using the Metropolis-Hastings algorithm.

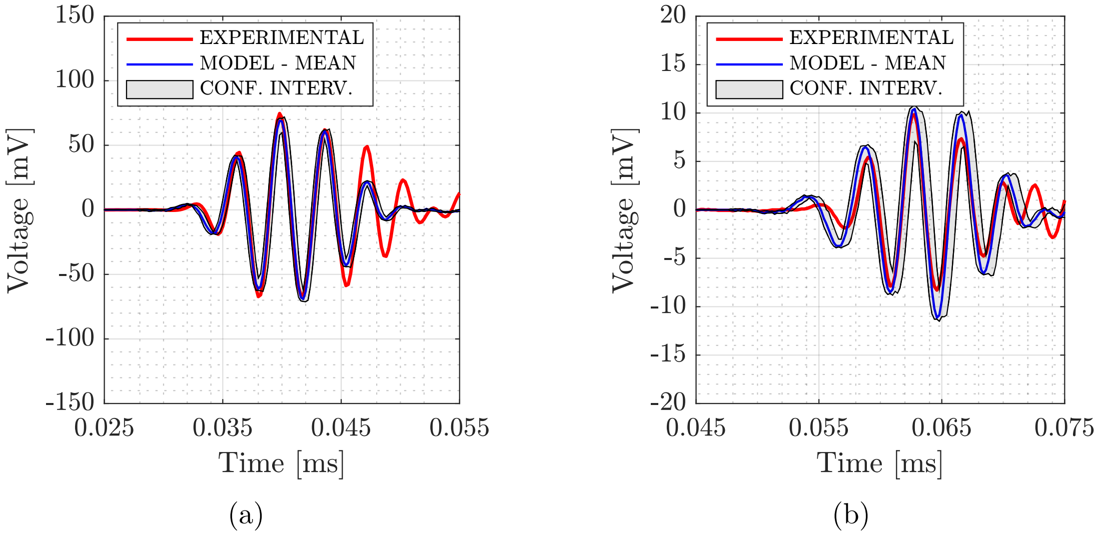


Source: Prepared by the author.

Figure 49 – Comparison between distributions obtained using the numerical and surrogate models at 20°C: (a) *a posteriori* distribution for E_1; (b) *a posteriori* distribution for G_{12}; and (c) two dimensional *a priori* and *a posteriori* distributions.

The FE and surrogate models are evaluated using the values within the 99% confidence interval as depicted in the histograms in Figs. 49a and 49b. This procedure yields the confidence interval for the output of the models. Figure 50 illustrates the confidence intervals for the signals obtained from the surrogate model. Observations reveal that the confidence interval for the signal measured at PZT-2 is narrower than that of PZT-3. This phenomenon can be attributed to the greater dispersion observed in the data obtained from PZT-3, as shown in Fig. 38 along with a lower signal-to-noise ratio.

Table 6 summarizes the results. Note that the surrogate model does not introduce significant differences in the Bayesian inference process when compared to the results obtained from the FE model. In addition to the results being similar, the proposed framework using the surrogate model is computationally efficient and significantly reduces



Source: Prepared by the author.

Figure 50 – Experimental (▬), ANN model mean (▬) and model confidence interval (■) signal Comparison for 20°C: (a) PZT 2; and (b) PZT 3.

the time required for analyzing the Markov chains. Excluding the initial parameter tuning phase, a complete Markov-chain Monte Carlo round with 1000 samples requires at least 32 hours when using the FE model. The initial tuning phase can take up to a working day, relying on multiple short simulations to analyze the behavior of the Markov chain and to adjust the hyperparameters. In contrast, the same algorithm implemented with the surrogate model takes about 240 seconds, which is almost 500 times faster than the FE model; additionally, the hyperparameter tuning phase can be completed in just a few minutes. Moreover, the surrogate model enables simulations of longer chains, leading to an increase in the number of points collected from the posterior probability distribution and enabling multiple tests to assess the convergence of the method. The surrogate model introduces a primary computational cost related to the data required for training. However, this expense can be efficiently addressed by distributing the training simulations across multiple machines, thus reducing the overall time required. In contrast, traditional MCMC simulations must be performed sequentially due to the sequential dependency of each chain step.

Multiple temperature

The same strategy used for 20°C is applied to fit the FE model to the experimental data collected at temperatures from 0 to 60°C. However, these adjustments are only made using the surrogate model. The Markov chain parameters are selected at each temperature to maintain an acceptance rate after convergence of the algorithm between 30% and 50%

Table 6 – Mechanical properties obtained for 20°C.

Parameter	**Results**	**Model**	
		FEM	ANN
E_1 [GPa]	Mean	61.25	61.24
	Standard deviation	0.29	0.23
	CV	0.47	0.37
	1% Percentile	60.26	60.10
	99% Percentile	61.97	61.62
G_{12} [GPa]	Mean	10.29	10.32
	Variance	0.14	0.09
	CV	1.41	0.9
	1% Percentile	9.92	10.11
	99% Percentile	10.52	10.47
	Time	115000 s	240 s

Source: Prepared by the author.

at the end of approximately 4000 simulations. This is possible with the surrogate model since each 1000 MCMC takes around 3 to 4 minutes. Table 7 presents the results for E_1 and G_{12}, and Fig. 51 shows the sampled posterior distributions.

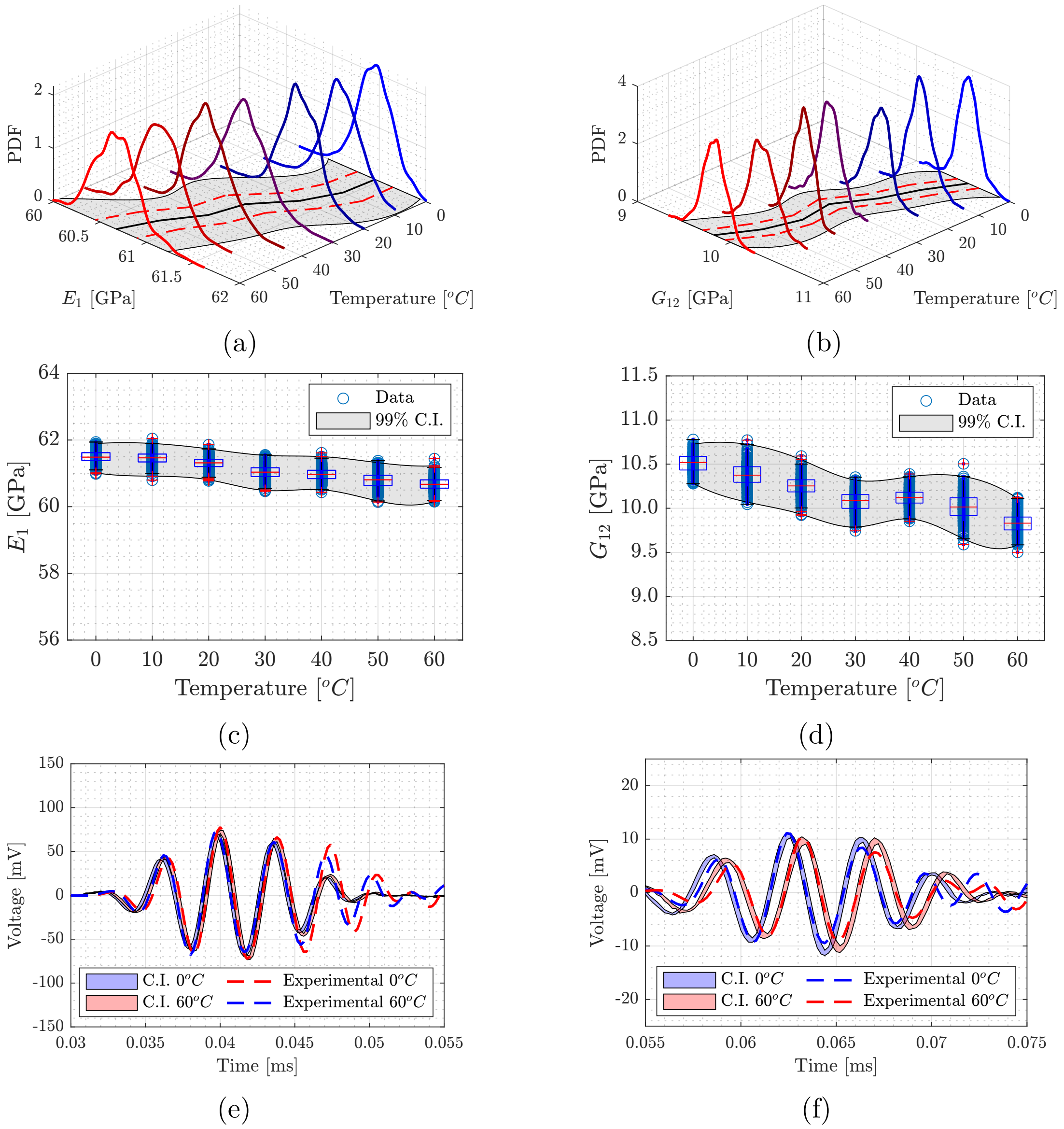
Table 7 – Mechanical properties obtained for 0°C to 60°C temperature range.

Parameter	**Metric**	**0°C**	**10°C**	**20°C**	**30°C**	**40°C**	**50°C**	**60°C**
E_1	Mean	61.46	61.44	61.30	61.06	61.00	60.80	60.67
	Std	0.191	0.207	0.190	0.215	0.195	0.214	0.225
	CV	0.310	0.337	0.310	0.352	0.320	0.351	0.371
	1% Percentile	60.94	60.91	60.82	60.55	60.57	60.20	60.08
	99% Percentile	61.98	61.05	61.83	61.67	61.51	61.34	60.32
G_{12}	Mean	10.56	10.38	10.26	10.06	10.11	10.01	9.82
	Std	0.116	0.124	0.108	0.114	0.103	0.131	0.121
	CV	1.09	1.19	1.05	1.14	1.02	1.30	1.23
	1% Percentile	10.14	10.06	10.01	9.74	9.84	9.70	9.50
	99% Percentile	10.87	10.69	10.55	10.38	10.32	10.40	10.11

Source: Prepared by the author.

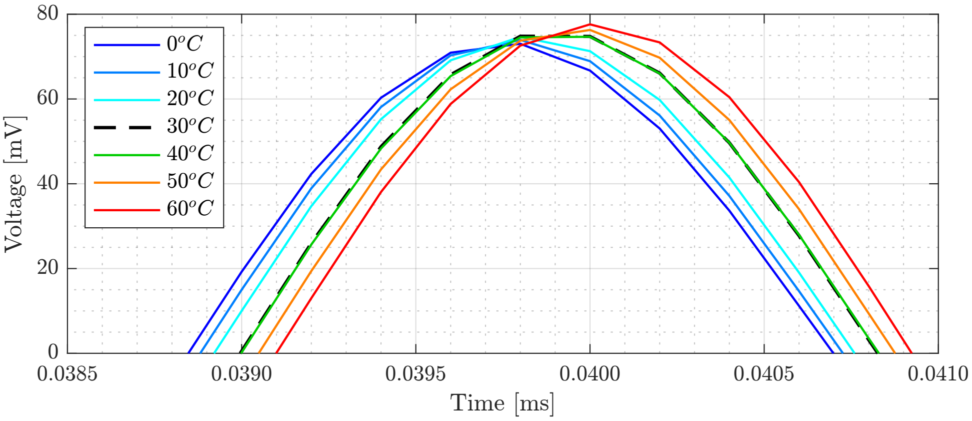
It is possible to notice a trend of reduction of material stiffness with increasing temperature, which is consistent with the behavior of a carbon/epoxy laminate. The trend of the mean results is mostly linear, as can be observed in Figs. 51c and 51d. The material's elastic modulus starts at 61.46 GPa at 0°C and decreases with temperature to reach the value of 60.67 GPa at 60°C, a variation of 1.3%. The shear modulus presents the same trend, from 10.56 GPa to 9.82 GPa, a variation of 7.0% . The more significant reduction on the shear module can be explained by the viscoelastic nature of the epoxy resin. This larger variation for the values of G_{12} compared to E_1 results in a greater dispersion of the signals for PZT3, which, situated 45 degrees from the excitation point, is more affected by the material's shear modulus.

The only exception to the observed trend in Figs. 51c and 51d is the value of G_{12} at a temperature of 30°C, which appears as an outlier, displaying a lower value than



Source: Prepared by the author.

Figure 51 – Confidence intervals for probability density functions for E_1 and G_{12} with temperature: (a) 3D distribution for E_1; (b) 3D distribution for G_{12}; (c) confidence intervals for E_1; (d) confidence intervals for G_{12}; (e) experimental results (- - -) and confidence intervals for signal of PZT 2; and (f) experimental results (- - -) and confidence intervals for signal of PZT 3. C.I. represents the confidence intervals.



Source: Prepared by the author.

Figure 52 – Close-up of the experimental signal wave peaks from PZT-2 in the temperature range of 0°C to 60°C.

that obtained at 40°C. To investigate this anomaly, the measured signals for PZT-2 were analyzed concerning temperature variations. Figure 52 shows a zoom near the peak of the first wave package from Fig. 38(c). The wave signals exhibited a consistent trend of delay with temperature increase, except for the measurement at 30°C, which deviated from the expected trend. Subsequent investigation into the experimental setup indicated a potential issue with the temperature controller in the thermal chamber on the day of the experiment. The ambient temperature fluctuated above 30°C, and the ON-OFF control logic of the controller faced challenges in maintaining an exact internal atmosphere of 30°C. Despite this experimental anomaly, the proposed fitting methodology demonstrated robustness. It effectively captured the temperature trend in the experimental data and accurately reflected it in the numerical model.

4.5 CONCLUDING REMARKS

This Chapter presents a framework that incorporates temperature effects in Lamb wave simulations within composite structures, in the context of SHM applications. Using Bayesian inference, a stochastic FE model was developed to accurately simulate Lamb wave behavior for different temperatures.

The framework consists of a three-step process: (i) initial adjustment, followed by (ii) a Sobol sensitivity analysis, and (iii) a stochastic adjustment through Bayesian inference and MCMC simulations. This framework has demonstrated its effectiveness in calibrating the FE model by considering uncertainties in the relevant mechanical parameters of the composite plate's fiber/resin system. The model results align closely with experimental observations within the considered confidence interval.

Incorporating temperature effects is a critical consideration while developing reliable SHM systems based on Lamb waves due to their sensitivity to environmental conditions. The proposed framework effectively incorporates these effects into the FE model, allowing it to simulate Lamb wave signals from 0 to 60 °C and to reproduce the experimental dispersion.

An additional contribution of this work is the integration of a machine learning-based surrogate model within the Bayesian inference process. Leveraging the capabilities of neural networks, the surrogate model significantly reduces the time required for MCMC rounds, thereby accelerating the posterior distribution sampling process. In the demonstration case, the time required for the MCMC rounds was reduced from approximately 30 hours using the classical FE model-based approach, to less than 5 minutes with the surrogate model approach. This time reduction is achieved without compromising the FE model's accuracy, ensuring the reliability of the results.

The integration of Bayesian inference and a machine learning surrogate model for model updating brings us one step closer to practical SHM implementations for composite materials. The proposed framework provides a foundation for reliable and accurate simulations and paves the way for constructing a robust SHM system suitable for practical, real-world implementation. Leveraging the stochastic model, Chapter 5 presents the application of this Bayesian updating framework to produce a hybrid dataset to train a 1D CNN to quantify damage in the CONCEPT dataset.

5 BAYESIAN DATA-DRIVEN FRAMEWORK FOR SHM UNDER LIMITED DATA

The reliability of a machine learning algorithm is limited by the amount and quality of the data utilized in its training stage. As presented in Chapter 3, the performance of such algorithms is heavly affected by the amount of data used in training phase. However, it is not feasible to intentionally damage most structures and measure their behavior under all potential damage scenarios and environmental conditions. Even in laboratory settings, creating specimens with multiple predetermined and characterized damage types and locations is a challenge. To overcome this, hybrid training strategies integrate supplementary data derived from physics-based restrictions and/or mathematical models into the training process of machine learning algorithms (XU *et al.*, 2023; ZHANG; SUN, 2020). Within this work, Chapter 4 proposes a Bayesian stochastic updating framework for FE models that simulate Lamb waves. These stochastic models are capable of reproducing the experimental results within a confidence interval, reducing the number of tests needed to characterize a structure.

This Chapter introduces a novel data-driven approach for monitoring composite structures using Lamb waves through a 1D CNN trained with data derived from a Bayesian-updated stochastic FE model. This approach utilizes data generated from an FE model, specifically calibrated for Lamb wave propagation using Bayesian inference, to train a 1D CNN for damage assessment in composite structures under varying temperatures. The Bayesian process refines the FE model using limited experimental data, employing a Markov-Chain Monte Carlo (MCMC) technique and the Metropolis-Hastings sampling algorithm. During this updating process, surrogate models based on MLP neural networks are used to reduce the computational demands of the MCMC simulations. The use of a stochastic FE model to generate training data presents several advantages: it lowers the experimental data requirements for developing the classifier, streamlines the training process, and potentially increases accuracy in data-limited situations by providing a more comprehensive dataset for classifier training.

The proposed training strategy combines the extensive generalization capabilities of CNNs with the statistical rigor of Bayesian inference in model updating, yielding a robust data source for training SHM machine learning algorithms. Incorporating data from the stochastically updated FE model into the training phase allows the 1D CNN to utilize the FE model's inherent uncertainty and extend its applicability to untested experimental data. This strategy enhances the classifier's ability to identify essential features from the FE model, thereby improving its predictive accuracy, especially when trained with a constrained experimental dataset.

The effectiveness of the suggested framework is evaluated across three distinct training contexts: exclusive experimental data, purely numerically generated data, and a combined set of experimental and numerical data. This part of the research proposes two significant contributions. First, it outlines the forging of a Bayesian data-driven machine-learning technique, drawing from a stochastic FE model, to quantify damage magnitude in composite structures under fluctuating temperatures. Second, it delves into the interplay between experimental and numerical data on the performance of a 1D CNN architecture regarding both confidence intervals and prediction quality.

The structure of this chapter is as follows. Section 5.1 provides an overview of the proposed framework, including the relevant theoretical background. Section 5.2 details the experimental setup, the FE model, the surrogate modeling strategy, and the datasets utilized during the 1D CNN training rounds. Subsequently, in Section 5.3, Bayesian inference is applied to update the FE model, the hyperparameters of the 1D CNN are tuned, and the proposed Bayesian data-driven machine learning strategy is evaluated in terms of accuracy and the impacts of temperature variations. Finally, Section 5.4 engages in discussions about the results, while Section 5.5 presents the final remarks and suggests potential directions for future research

The work presented in this chapter is based on the article "Bayesian data-driven framework for structural health monitoring of composite structures under limited experimental data", published in the Structural Health Monitoring journal.

5.1 PROPOSED FRAMEWORK

This Section describes the main parts of the proposed Bayesian data-driven framework to quantify damage size on a composite structure. The proposed framework consists of two parts i) stochastic FE model development and ii) 1D CNN training and evaluation. Fig. 53 presents the proposed framework used for this work. The first step comprises a stochastic updating of a FE model using Bayesian inference through MCMC simulations with the Metropolis-Hastings sampling algorithm. A sensitivity analysis using Sobol indices and a surrogate modeling strategy defines the relevant parameters and speeds up the MCMC rounds, respectively. Following the updating process, a 1D CNN is trained with datasets from three different strategies: (i) only experimental data, (ii) only numerical data, and (iii) a hybrid combination of both.

The proposed framework is grounded on the following assumptions and considerations:

1. The experimental dataset made available by da Silva et al. (SILVA *et al.*, 2020) is used as an experimental setup, and it is considered that no material property is known *a priori*.

2. The experimental database is divided into training and test datasets. The test dataset is used only at the end to evaluate the performance of the machine learning algorithms.

3. The training dataset is used to update the material properties of a FE model.

4. The FE model accounts for the experimental results variability through variations in material properties; therefore, there is not a unique value for a given property. Instead, there is a Probability Density Function (PDF) for each significant parameter.

5. A Bayesian inference process using MCMC simulations and the Metropolis-Hasting sampling algorithm is effective in sampling posterior PDFs from uniform prior distributions.

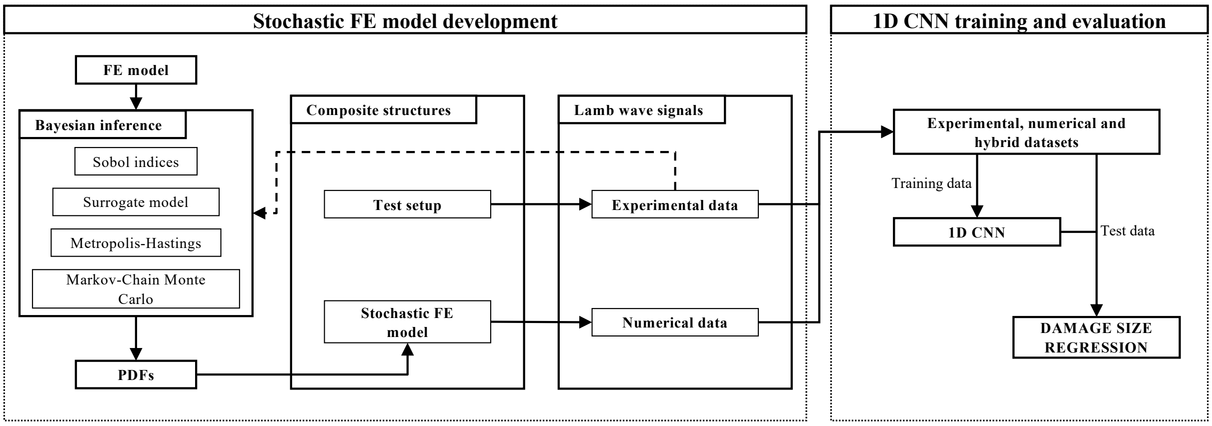
6. The FE model can be used to simulate other conditions inside and outside the range of damages in which it is calibrated.

7. A surrogate model, based on neural networks, can replace the FE model to reduce the time required for the MCMC rounds.

8. A 1D CNN can gather features of the raw data series. It can be trained using experimental data, numerical data, or a hybrid combination of both, comprising three different types of datasets.

9. The proposed strategy is evaluated by comparing the performance of machine learning algorithms trained in the three types of datasets.

10. The network performance is evaluated using a test dataset not used during the Bayesian inference process or the training stage.

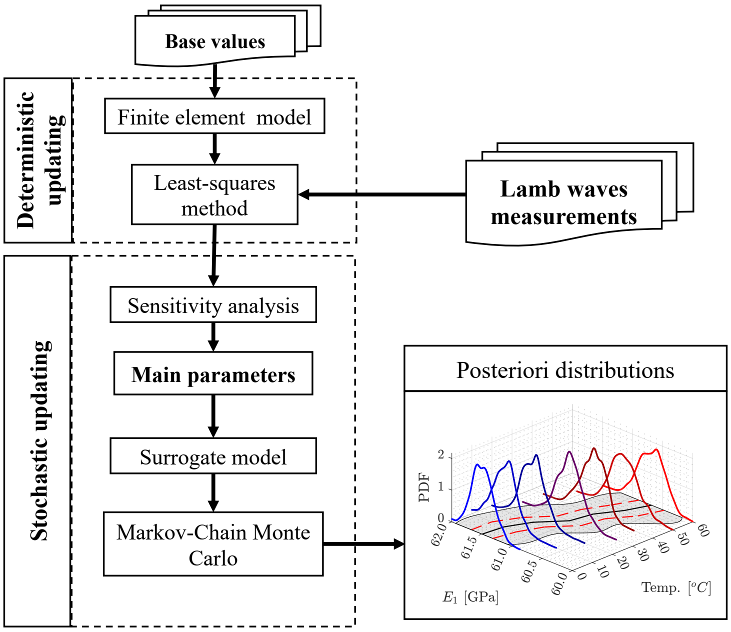


Source: Prepared by the author.

Figure 53 – Proposed Bayesian-driven framework for stochastic FE model updating, data generation, and machine learning algorithm training and evaluation.

5.1.1 Stochastic FE model development

This work uses the Bayesian inference process proposed in Chapter 4 to obtain a stochastic FE model for Lamb wave propagation under varying temperatures. Figure 54 presents the proposed strategy, divided into two main steps: an initial deterministic updating step and a stochastic updating procedure. The deterministic updating is based on a variation of the least-squares method. In contrast, the stochastic updating procedure is composed of a sensitivity analysis using Sobol indices followed by an MCMC simulation using the Metropolis-Hastings sampling algorithm to sample the posterior probability function of the main parameters of the model, i.e., those that influence the model output the most. The framework is applied for multiple temperatures, enabling the construction of a model capable of emulating a complete experimental dataset across a range of temperatures. Section 4.2 discusses in details the framework and the theory background.



Source: Prepared by the author.

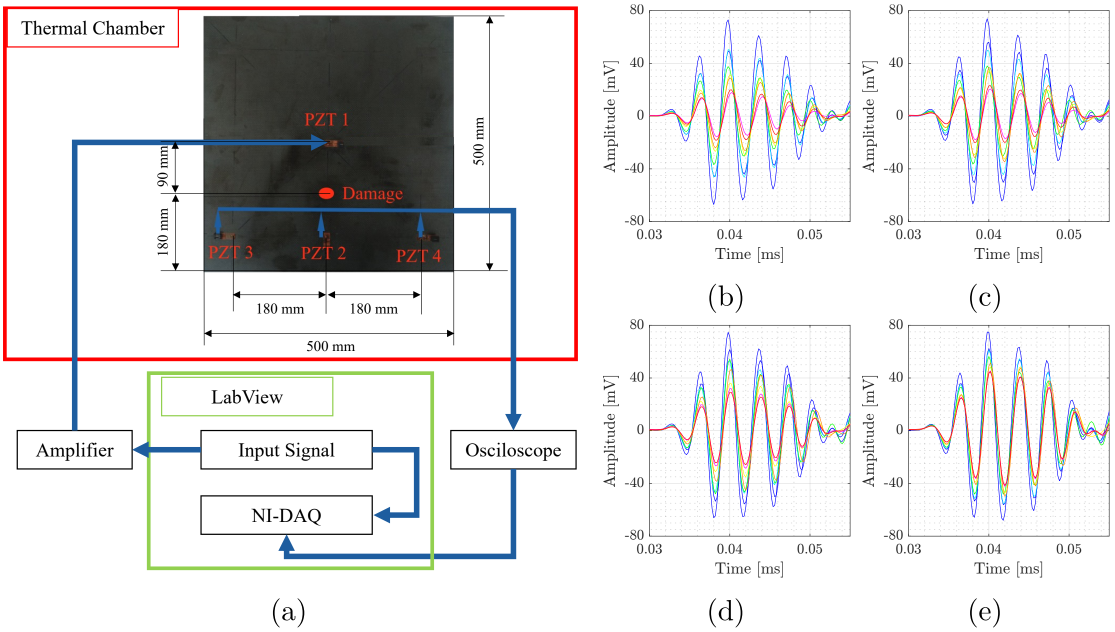
Figure 54 – Stochastic FE model updating process.

5.2 EXPERIMENTAL SETUP

The experimental data available in the CONCEPT dataset (refer to Section 4.3.1) is used in this work. The dataset consists of a carbon fiber reinforced polymer (CFRP) plate with dimensions of 500 x 500 x 2 mm and made of 10 layers of plain weave fibers oriented along the plate borders directions. To detect the behavior of the plate, four PZT SMART Layers sensors with a diameter of 6.35 mm are bonded to the plate with the help of epoxy resin. The data acquisition is conducted under controlled temperatures, which are incrementally increased in steps of 10°C from 0°C to 60°C. Section 4.3.1 presents further details about the data acquisiton process and Fig. 55a depicts the experimental setup. The dataset is publicly available in the GitHub[1] repository CONCEPT: CarbON-epoxy CompositE PlaTe (SILVA *et al.*, 2020).

In addition to the undamaged conditions detailed in Secion 4.3.1, damaged conditions are simulated by adding mass to the plate, with increasing diameter, ranging from 20 to 80 mm in increments of 10 mm. This mass addition simulates local variation on the damping of the plate (LEE *et al.*, 2011) and allows reversibly simulating damage. Overall, the dataset comprises 28 damaged conditions. Figures 55b to 55e show one experimental measurement for each temperature from 0°C to 60°C and diameter from 20 to 80 mm. It is important to emphasize that the impact of damage is significantly influenced by

[1] https://github.com/shm-unesp/DATASET_PLATEUN01



(a) (b) (c) (d) (e)

Source: Plate image adapted from Silva *et al.* (2020).

Figure 55 – Experimental setup: (a) data acquisition strategy; and results for progressive damaged conditions from 20 mm to 80 mm at (b) 0°C; (c) 10°C; (d) 30°C; and (e) 60°C. In the charts (b) to (d), the colors indicate progressive damage from baseline (blue) to 80 mm damage (red).

temperature, primarily due to the characteristics of the material employed for simulating the damage.

In the current study, the database being analyzed, encompassing a temperature range of 0°C to 60°C and damage diameters of 20 to 80 mm, consists of a relatively small sample size, with only 28 data points across these dimensions (see Tab. 8). This limited sample size poses a challenge in splitting the data into conventional training, validation, and testing sets, typically distributed in ratios such as 50/25/25, 70/15/15, or 80/10/10. Adopting such divisions would result in validation and test subsets with merely 3 or 4 samples each, insufficient for effectively evaluating machine learning algorithms both during training and testing phases.

To address this issue, this work explores the potential benefits of supplementing the sparse experimental data with additional data generated through a stochastic FE model. This approach aims to enrich the sample space, thereby overcoming the limitations posed by the small experimental dataset. The proposed Bayesian framework plays an important role in this context, as it facilitates the generation of synthetic data that includes the experimental uncertainty inside the model's confidence interval. By using this framework, the study aims to evaluate the advantages of using a combination of real and synthetic data, particularly in scenarios where experimental data is limited.

Therefore, to comprehensively evaluate the performance of the proposed Bayesian framework, the experimental data is arranged into a scenario in which data could be feasibly collected at various temperatures with minimal cost in a practical scenario, as depicted in Tab. 8. This division simulates a common laboratory situation in which the structure could be damaged in limited conditions and tested at several temperatures. This database allows for the evaluation of temperature interpolation capacity in two ranges with different sizes - from 10 to 30°C and from 30 to 60°C, as well as the analysis of damage interpolation capacity between 30 and 70 mm, and the study of the extrapolation capacity with damages of 20 and 80 mm.

Table 8 – Data division between training (🟩), validation (🟦), and test (🟥).

Damage diameter	**Temperature [°C]**			
[mm]	0	10	30	60
20	🟥	🟥	🟥	🟥
30	🟩	🟦	🟩	🟦
40	🟥	🟥	🟥	🟥
50	🟦	🟩	🟦	🟩
60	🟥	🟥	🟥	🟥
70	🟩	🟦	🟩	🟦
80	🟥	🟥	🟥	🟥

Source: Prepared by the author.

5.2.1 FE model

An ABAQUS/Explicit FE model similar to the described in Section 4.3.2 is used to simulate the plate, with the addition of damage. The plate is modeled using continuum shell elements (SCR8). These elements are designed to discretize a three-dimensional body rather than just representing a surface like traditional shell elements. Continuum shells solely possess displacement degrees of freedom, adopt linear interpolation, and account for the impact of transverse shear deformation and thickness alteration. Furthermore, they are founded on first-order composite theory and can be stacked to enhance the precision of the through-thickness response.

The analysis's spatial and temporal resolution influences the numerical simulation's stability. Moser (MOSER; JACOBS; QU, 1999) suggests that a spatial resolution of no less than 20 nodes per wavelength is essential to guarantee stability, which can be expressed as $l_e = \lambda_{min}/20$, where λ_{min} is the minimum wavelength anticipated in the model and l_e is the characteristic length of the mesh. Furthermore, to avert numerical instability, the time step must be determined based on the highest anticipated frequency in the simulation, with a suggested value of $\Delta t = 1/(20 f_{max})$, where Δt represents the time step and f_{max} denotes the maximum frequency expected in the model. When the input frequency is 250 kHz, adhering to both considerations leads to a mesh comprising element sizes of approximately 2 mm and a maximum time increment of 0.2 μs. However, further stability analyses of the model reveal that a time increment of 0.05 μs is required to address mesh irregularity surrounding the PZT regions, guarantee convergence with the presence of damping induced by the damage mass, and maintain a reasonable simulation time.

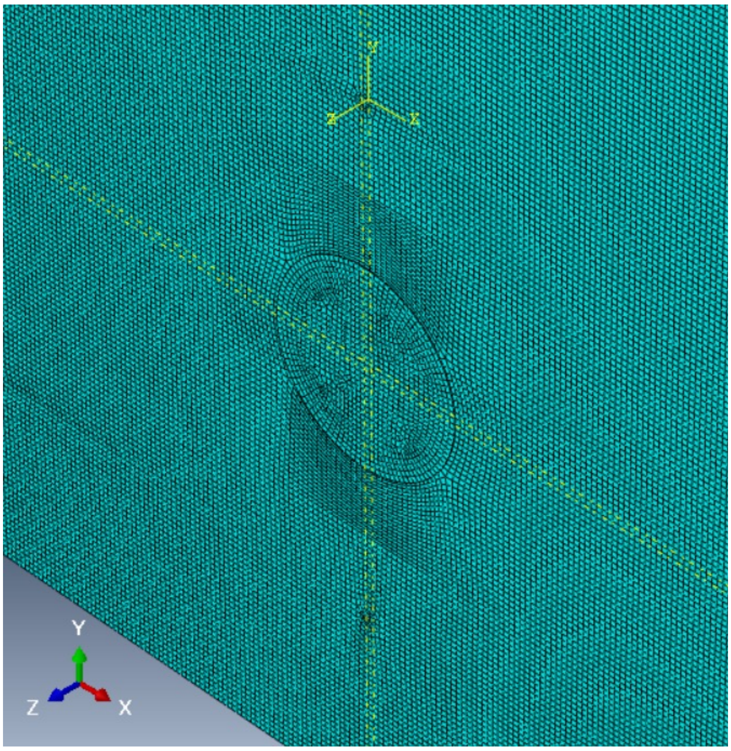
Figure 39 presents the geometry of the FE model. To facilitate the generation of a structured mesh for the entire plate, the plate is divided into several smaller rectangular regions. The area where the PZT actuators and sensors are attached to the plate is modeled explicitly with a circular mesh. As proposed by Gresil et al. (GRESIL *et al.*, 2013), the excitation signal is applied as eight self-balancing forces around this region, as depicted in Fig. 39b. To obtain the output signal, the integrated strain results within the sensor region are used and transformed to voltage as suggested by Sirohi, and Chopra (SIROHI; CHOPRA, 2000). The region between PZTs 1 and 2 is modeled with a circular mesh to account for damage addition. Damage is modeled as a constant thickness circular disk with solid elements and a tie interaction with the plate.

The composite material is implemented in the FE model using ply-based properties. The mechanical behavior of the materials is modeled using the classical laminate theory, where each layer is considered an orthotropic material with 3 integration points per lamina. As such, Young's modulus in two perpendicular directions in the plane (E_1 and E_2), Poisson's ratio (ν_{12}), shear moduli in and out of the plane (G_{12} and G_{23}, respectively), and density (ρ) had to be determined. As a simplification, it is assumed that the values of E_1 and E_2 for plain weave fibers are close, so they are considered equal. The emulated damage mass (see Fig. 56) is modeled as an elastic material with Rayleigh damping, characterized by two coefficients, α and β (Eq. 38). It can be observed that in Rayleigh's damping, α predominates at low frequencies, while β dampens high frequencies

$$\xi = \frac{\alpha_R}{2\omega} + \frac{\beta_R \omega}{2}. \tag{38}$$

Figure 56 presents a close-up view of the mesh around the damage.

The model accounts for temperature effects at the material level. Initially, the elastic modulus in perpendicular directions (E_1 and E_2), Poisson's ratio (ν_{12}), and shear moduli in and out of the plane (G_{12} and G_{23}) are heuristically selected as factors affected by temperature for the composite material. The Young modulus, Poisson coefficient, and



Source: Prepared by the author.

Figure 56 – Detail of the implementation of the simulated damage in the FE model.

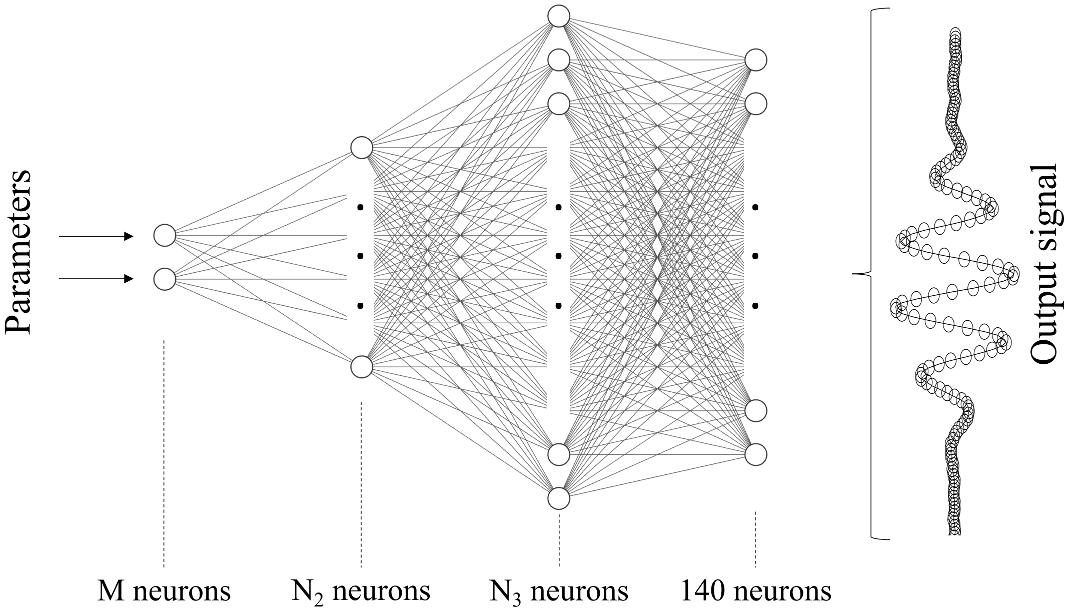
Rayleigh damping coefficients (α and β) are considered temperature dependent for the damage mass simulation material. The specific mass of both materials is considered constant with temperature but treated as an uncertain quantity for both materials.

The series of Python routines presented in Fig. 40 are updated to include the damage parameters, and the FE model is controled through the MATLAB command line interface.

5.2.2 Neural network surrogate model

To accelerate the time-consuming MCMC simulations, the authors propose employing a surrogate model, similar to the one developed in Section 4.4.3, to replace the FE-based model during the random-walking process. In this strategy, the FE model is extensively simulated with properties from a bounded uniform distribution to generate the necessary training data. The boundaries are defined by the results of the deterministic updating procedure, which is increased by a margin of 5-20%, depending on the uncertainty about the parameter, to allow a random-walk margin for the MCMC algorithm. The model response is truncated after the first wave packet to reduce training time and data requirements for network convergence. The neural network is implemented using the MATLAB deep learning toolbox and is trained using the Adam algorithm. The training process employs a dataset division with 80/10/10 proportions for the training, validation, and testing sets, respectively, with the loss on the validation data serving as stop criterion for the training process. The input variables are normalized between 0 and 1 to enhance convergence, while the output signal is normalized between -1 and 1.

The input layer has M neurons representing the random variables updated within the Bayesian inference process, and the output layer contains 140 neurons representing the time series output. The number of hidden layers is defined as two, with the number of neurons N_2 and N_3 at the first and second hidden layers selected through exploratory search. Figure 57 depicts the surrogate model structure.



Source: Prepared by the author.

Figure 57 – Architecture of the MLP used as surrogate-model.

5.3 RESULTS

The results are divided into two subsections (i) the stochastic FE model updating results and (ii) the training and evaluation of the machine learning model. The former is further divided into undamaged and damaged model updating, while the latter is divided into 1D CNN structure definition, damage quantification, and temperature effects.

5.3.1 Stochastic FE model updating

The updating procedure detailed in Section 5.1.1 is applied to the FE model in the undamaged and damaged conditions. At the undamaged condition, only the plate material parameters are updated (E_1, E_2, ν_{12}, G_{12}, and G_{23}) using experimental data ranging from 0 to 60°C. This Section briefly recalls the results from Section 4.4.2. Conversely, for the damage condition, the emulated damage mass material is updated (E, ν, α, and β) using the available information for 0, 10, 30, and 60°C. Also, the updating procedure of the damage condition only used the experimental signals for diameters of 30, 50, and 70 mm. The other damage diameters are used to evaluate the performance of the proposed damage quantification strategy at the testing phase.

Undamaged model updating

For the undamaged condition, it is necessary to establish an initial search range for the design parameters space. Due to intellectual property agreements, the available information about the plate is limited to its composition, which consists of aerospace-grade carbon fiber with 10 layers in a plain weave configuration (FRANCA, 2014). This range is determined based on information from the literature for plain weave carbon/epoxy systems, with the main source being the work of Paiva et al. (PAIVA; MAYER; REZENDE, 2006).

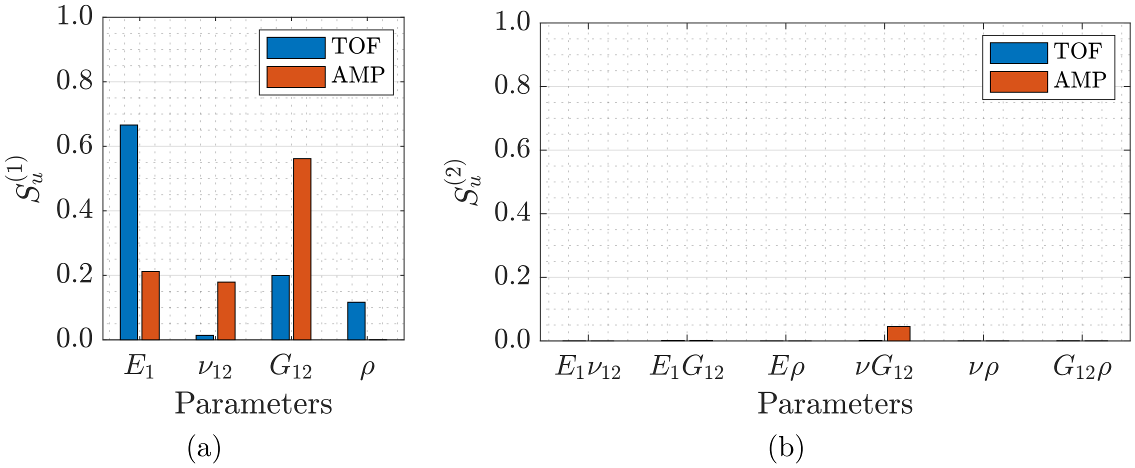
The material properties are initially updated using the iterative process based on the modified least-squares method described in Section 5.1.1. Table 9 presents the property limits and the initial and optimal values.

Table 9 – Deterministic updating procedure results

Parameter	**Min**	**Max**	**Initial**	**Optimal**
E_1 [GPa]	60.0	80.0	70.0	61.2
E_2 [GPa]	60.0	80.0	70.0	61.2
ν_{12}	0.05	0.15	0.10	0.075
ρ $[Kg/m^3]$	1550	2250	1800	1620
G_{12} [GPa]	5.00	15.0	7.50	10.25
G_{23} [GPa]	4.00	6.00	5.00	5.00

Source: Prepared by the author.

A sensitivity analysis using Sobol indices is conducted to identify the key factors influencing the model's response. All properties are randomly perturbed around their optimal value, and the model's sensitivity is evaluated accordingly using the metrics depicted in Fig. 42a. The results for the first-order Sobol indices are presented in Fig. 58. Only E_1 and G_{12} significantly impacted the model's response in the undamaged model. E_1 corresponds to approximately 65% of the variance on TOF, complemented by 20% from G_{12}, while 55% of the variance on the amplitude of the signal is due to G_{12} complemented by 20% from E_1 influence. The remaining parameters represent less than 20% of the variance, and the second-order Sobol indices show that the influence of combinations of parameters is negligible. Therefore, ν_{12} and ρ are assigned fixed values of 0.075 and 1620 kg/m^3, respectively.



Source: Prepared by the author.

Figure 58 – Sobol indices for the undamaged model: (a) first; and (b) second order. In the figure legends, TOF and AMP mean time of flight and amplitude, respectively.

The remaining parameters are undetermined, and it is necessary to update their posterior probability distributions and gather information from experimental data containing environmental variation. In summary, even small inherent variations associated with uncertainties in the experimental data can lead to slight changes in the response pattern under the same test conditions, thereby influencing the identified values of these parameters. Hence, it is evident that deterministic estimation of these parameters cannot be universally applied in the presence of experimental data uncertainties; instead, they must be treated as random variables.

The proposed methodology for Bayesian inference presented in Section 5.1.1 is applied to the experimental data. The variables $\Theta = \{E_1, G_{12}\}^T$ had central values defined as the optimal obtained from the deterministic updating, and the random walk algorithm searched for possible candidates around these values.

Due to the significant time requirements of MCMC simulations, it is proposed to utilize a surrogate model based on neural networks to replace the FE model in the context of random walking. The FE model is extensively simulated, employing randomly sampled properties from a bounded uniform distribution. Specifically, the parameters E_1 and G_{12} are sampled from the uniform distributions $\mathcal{U}(54, 66)$ [MPa] and $\mathcal{U}(8.5, 11.5)$ [MPa], respectively. These bounds encompass the deterministic values while allowing for adequate flexibility to accommodate variations in the physical properties due to temperature changes. The resulting data from these simulations are utilized for training a surrogate model as described in Section 4.4.3. To determine the number of neurons N_2 and N_3, the surrogate model is simulated with varying numbers of neurons ranging from 50 to 300, and the results are evaluated using the root mean error between the FE-model results and ANN results. An error convergence analysis demonstrated that a network with 50 neurons in the second layer and 200 neurons in the third layer trained with 500 samples could represent the FE-model results with less than 1% error.

Each MCMC round utilizing the FE model necessitates approximately 30 hours of computer time and must be executed on a single machine due to the sequential nature of Markov Chains. In contrast, generating 500 random samples for surrogate model training takes less than 20 hours and can be parallelized across multiple machines. Subsequently, employing the ANN surrogate model for an MCMC round takes around 5 minutes, and this same surrogate model can be utilized for multiple temperatures, provided that the properties' search space lies within the training limits.

For the Markov Chain estimation, the variance of the likelihood function is defined iteratively through rounds of the fitting algorithm, trying to ensure a stable acceptance rate between 40 and 50%. The chain is simulated over 4000 samples, and the first 20% samples are discarded (burn-in). Figures 60a to 60h present the statistical properties of the obtained properties and Figs. 61a to 61d depict the experimental signals compared to the model results and confidence intervals. The confidence intervals are obtained by evaluating the FE model within the 99% percentile for the input parameters.

Damaged model updating

Evaluating Eq. 38, one can notice that α is responsible for the damping component at low frequencies. Among the available parameters for adjusting the damaged model, initial tests showed that this damping component is insignificant to the model's response. As Lamb waves occur in ultrasonic ranges, specifically 250 kHz for the present study, only β is relevant. Therefore, the damaged model has four adjustable parameters besides the ones already considered in the undamaged model: E, ν, β, and ρ for the damage mass. The initial value of the mass density was determined experimentally as 1750 kg/m^3 and kept constant with temperature. An initial estimate is based on an exploratory search for the other parameters. Table 10 shows the proposed range of values for the parameters of interest. Note that, due to the uncertainty involved in the system, E and β have a wide search range. The Poisson's ratio is limited to 0.45 to avoid potential issues with incompressibility in the FE model.

Table 10 – Search region for the damaged model parameters.

Parameter	**Minimum value**	**Maximum value**
E [MPa]	100	700
ν	0.35	0.45
1e-7 β	0.5	10
ρ [kg/m^3]	1650	1850

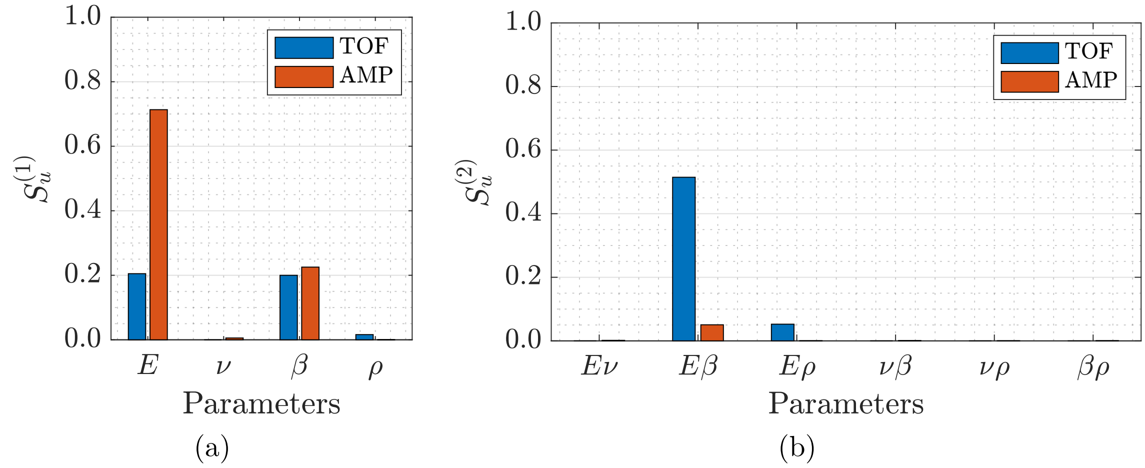
Source: Prepared by the author.

The updating procedure requires simulating the model with the three training diameters from Tab. 8 (30, 50, and 70 mm) at each parameter alteration. Moreover, adding damping to the system reduces the minimum step required for numerical stability to 0.05 ns. Consequently, the wide range of potential values, combined with multiple damage diameters, significantly increases the computational burden of employing the FE model in an exploratory search strategy. Therefore, an initial sensitivity analysis employing Sobol indices is used. A Latin hypercube sampling strategy with 200 samples is constructed, encompassing the four parameters from Tab. 10 and considering the fixed central damage diameter value of 50 mm. The 30 and 70 damage diameters are not included to keep the model's variance only related to the material properties.

Figure 59 presents the obtained first and second-order Sobol indices. Note that E represents almost 70% of the variance related to the signal's amplitude, complemented by 25% from β. Conversely, the variance of the TOF metric cannot be entirely addressed by individual parameters, as the second-order Sobol indices show that the combination $E\beta$ is responsible for half of this metric variance. The remaining parameters have negligible impact on the model's response; therefore, they are subsequently assigned fixed values, namely 0.4 for the Poisson's ratio and 1750 kg/m^3 for the mass density. E and β are considered undetermined, and updating their posterior probability distributions is necessary following the Bayesian inference procedure described in Section 5.1.1.

To perform the MCMC random walks, a surrogate model is constructed using 1000 samples obtained from the FE model, which was simulated using the uncertain properties sampled from a Latin hypercube bounded by $E \sim \big(100, 700\big)$ [MPa] and $\beta \sim 1e^{-7}\big(0.5, 10\big)$, along with damage diameters of 30, 50, and 70 mm. The ANN surrogate model underwent the Bayesian inference process using MCMC with the Metropolis-Hastings algorithm. For the Markov Chain estimation, the variance of the likelihood function is defined iteratively through rounds of the fitting algorithm, trying to ensure a stable acceptance rate between 40 and 50%. The chain is simulated over 4000 samples, and the first 20% samples are discarded (burn-in).

Figs. 60a to 60p present the sampled PDFs for E_1, G_{12}, E and β. The PDF is obtained



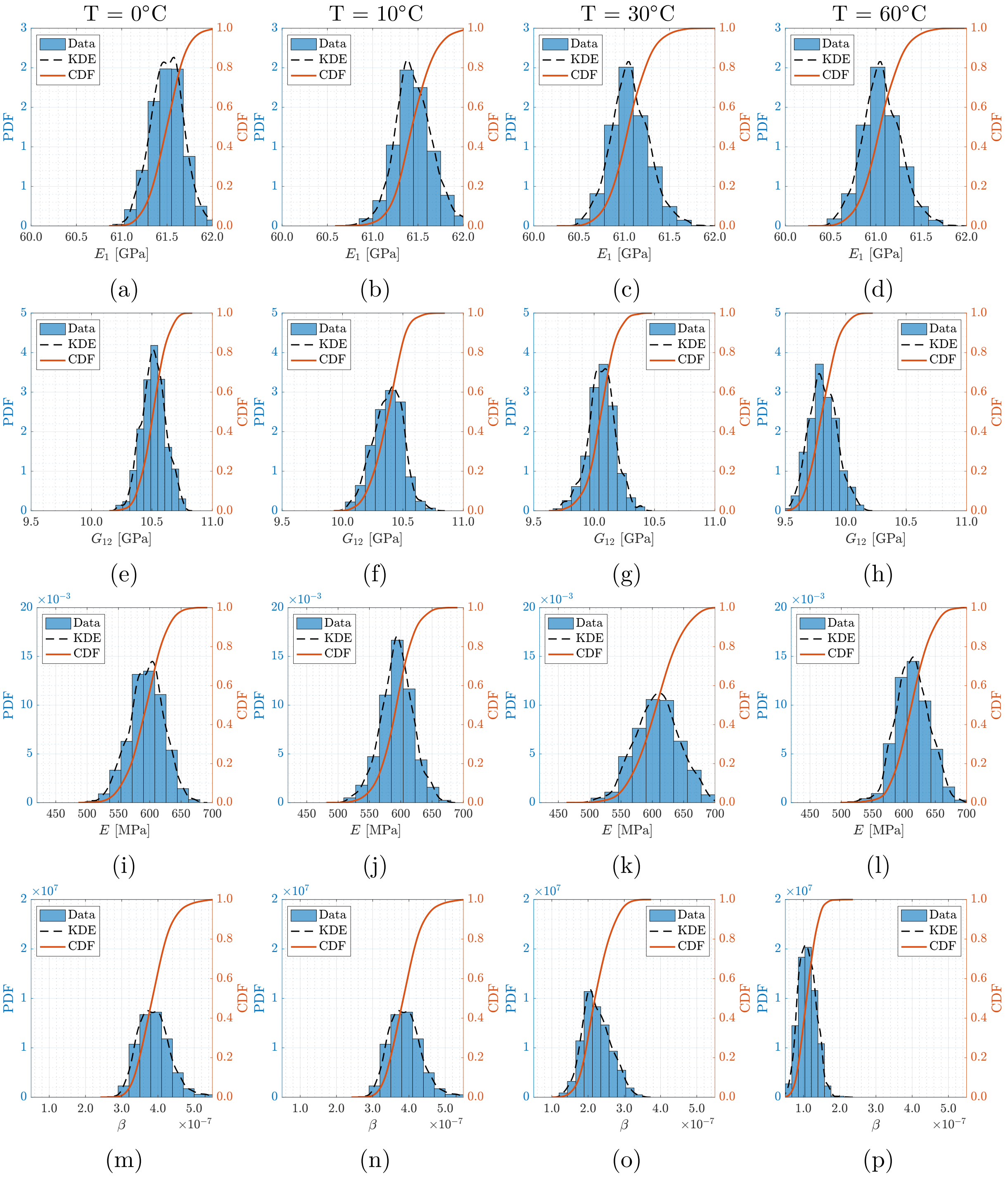
Source: Prepared by the author.

Figure 59 – Sobol indices for the damaged model: (a) first; and (b) second order. In the figure legends, TOF and AMP means time of flight and amplitude, respectively.)

from a kernel density estimate (KDE) applied to the MCMC sampled data. Note that all material properties tend to reduce their magnitude with temperature. Comparing the CFRP and the emulated damage mass, the latter has a greater variation, with Young modulus and β damping coefficient average values varying from 575 MPa and $3.9{\times}10^{-7}$ to 475 MPa and $1.5{\times}10^{-7}$, respectively.

Samples obtained using the MCMC process are considered a set that approximates the target distribution. These obtained samples are then used as a surrogate PDF representing the target distribution and sample. Simulating the model within its 99% confidence interval for the sampled properties gives the model confidence intervals for the signals, as depicted in Figs. 61a to 61d.

The application of the framework resulted in a Bayesian updated FE model that can be simulated over all ranges of temperatures and provide results with variability within a confidence interval that encompasses the experimental results. The model is a foundation for emulating the experimental setup with and without damage. During the data generation phase, given a temperature, the properties can be sampled from the obtained PDFs. Then, the FE model can be evaluated with the intended damage diameter.



(a) (b) (c) (d)

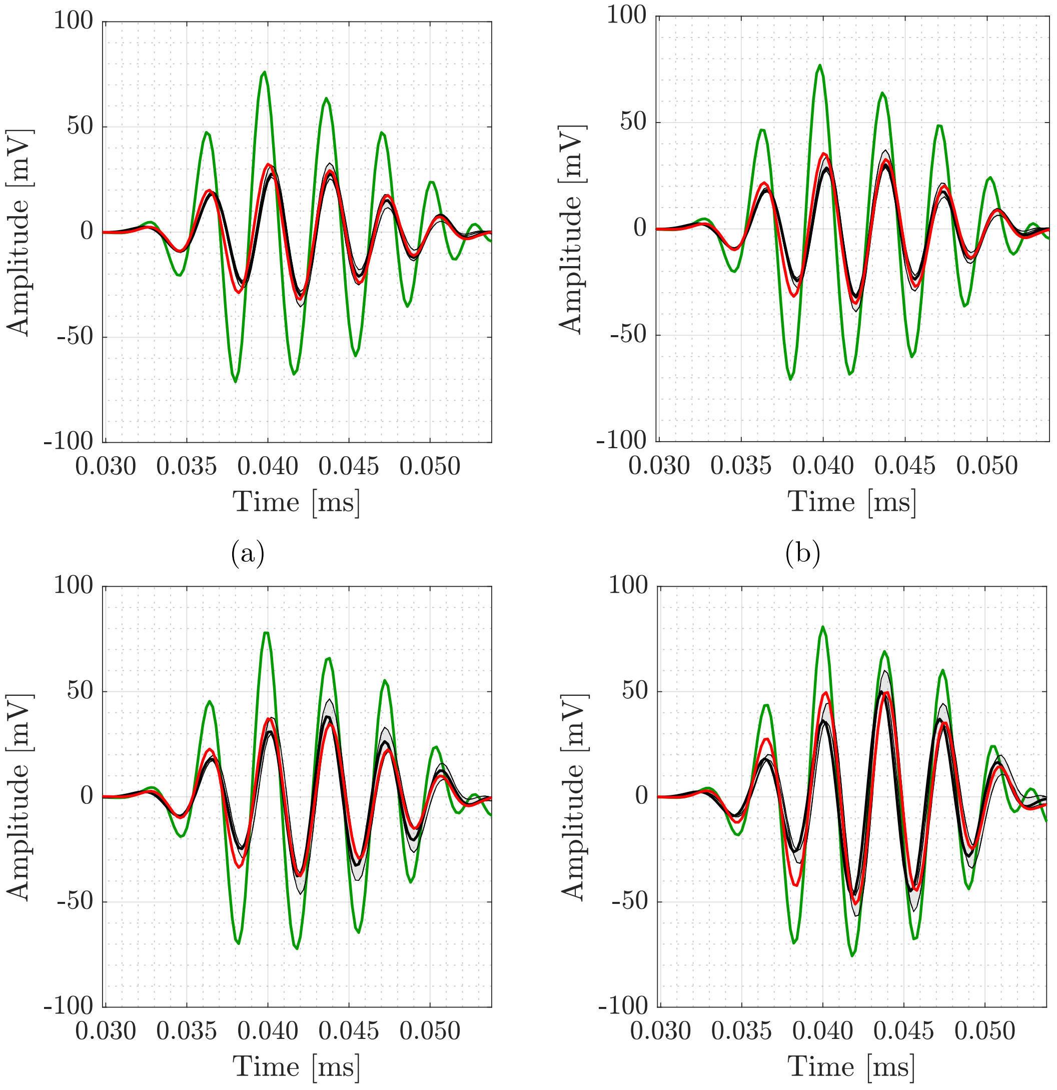
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Source: Prepared by the author.

Figure 60 – Results for stochastic updating procedure using 30, 50, and 70 mm damage diameters at 0, 10, 30, and 60°C, respectively, in the horizontal direction. (a) to (d) E_1 distribution; (e) to (h) G_{12} distribution; (i) to (l) E distribution; and (m) to (p) β bistribution. In the histograms, the colors represent: Sampled data (▮), Kernel Density Estimate (KDE) (–) and Cumulative Density Function (CDF) (–).



(a)

(b)

(c)

(d)

Source: Prepared by the author.

Figure 61 – Model response for PZT at (a) 0 °C; (b) 10°C; (c) 30°C; and (d) 60°C. In the graphs, the colors represent experimental undamaged signal (—), experimental 50 mm damaged signal (—), model prediction for 50 mm with mean properties (—) and confidence interval (▬).

5.3.2 1D CNN evaluation.

In the following subsections, the impact of data generated by the stochastic FE model on the proposed 1D CNN classifier is evaluated using three distinct strategies: (i) using only experimental data, termed EXP; (ii) using only numerical data, referred to as NUM; and (iii) employing a combination of both types of data, known as hybrid training or HYB. The strategy for preparing each dataset is detailed in Section 5.3.3, the model's structure is outlined in Section 5.3.4, and the data division strategies are assessed in terms of the model's accuracy and the temperature effect in Sections 5.3.5 and 5.3.6, respectively.

5.3.3 Datasets preparation

The experimental and numerical results must be pre-processed to generate the EXP, NUM, and HYB datasets. The first pre-processing step consists of selecting the first wave package of each k PZT in a damaged condition and concatenating them into a vector $\mathbb{R}^{1\times kM}$. M is defined as 140 points encompassing the FE model's validation region, and k is equal to 3. Therefore, the input signal has a length of $kM = 360$. Then, a reference undamaged signal with dimension $\mathbb{R}^{1\times kM}$ is paired with the damaged signal, creating a sample $\boldsymbol{x}^{(l)} \in \mathbb{R}^{2\times kM}$. Finally, each sample is paired with the scalar damage diameter, also known as severity, represented by $\boldsymbol{y}^{(l)} \in \mathbb{R}^{1\times 1}$ and normalized between 0 and 1, considering the range of 0 to 80 mm. The 1D CNN is trained to receive the $\boldsymbol{x}^{(l)}$ array and determine damage diameter $\boldsymbol{y}^{(l)}$ related to it.

The training is performed with the EXP, NUM, and HYB datasets. The EXP dataset is composed only of experimental data. There are 100 experimental signals for each temperature and damage condition of Tab. 8, and the damaged conditions are divided into training, validation, and test subsets according to it. The undamaged signals for each temperature are divided using the same proportion of the damaged conditions, i.e., 2/1/4 and 1/2/4 for 0 and 10°C, respectively. Finally, each damaged signal is permuted with all undamaged signals, expanding the available combination of signals for training, validation, and testing. Table 11 summarizes the data division.

This data partitioning strategy is chosen for two primary reasons: (i) Segregating undamaged from damaged signals before distributing them into training, validation, and test datasets ensures entirely distinct subsets. This means no overlap between these subsets, ensuring the neural network is not exposed to test samples during training. (ii) Given experimental uncertainties, two recorded signals under identical conditions are

Table 11 – Experimental and FE model data division and number of samples. The abreviations mean: Temperature - T; Undamaged samples - U; Damaged samples - D; Possible combinations - C; Training - Tr.; Validation - Val.; S - Set; and Test - Te.

	T	**0°C**			**10°C**			**30°C**			**60°C**		
Experimental	U	100			100			100			100		
	D	700			700			700			700		
	S	Tr.	Val.	Te.	Tr.	Val.	Te.	Tr.	Val.	Te.	Tr.	Val.	Te.
	U	30	15	55	15	30	55	30	15	55	15	30	55
	D	200	100	400	100	200	400	200	100	400	100	200	400
	C	6000	1500	22000	1500	6000	22000	6000	1500	22000	1500	6000	22000
FE model	U	50			50			50			50		
	D	500			500			500			500		
	C	25000			25000			25000			25000		

Source: Prepared by the author.

never the same. Consequently, an undamaged baseline can be contrasted with all damaged signals for a specific temperature. By shuffling the baseline within the same temperature bracket, we can diversify available samples for training. For the creation of the NUM and HYB datasets, the numerical signals are procured from the stochastic FE model simulations at temperatures of 0, 10, 30, and 60°C. Diameters are randomly chosen within the range of 20 to 80 mm. Undamaged signals are sourced by sampling PDFs for 50 times per temperature for the undamaged model. 500 diameters are randomly chosen within the 20 to 80 mm range for damaged signals at a specific temperature. PDFs of the properties are then sampled for each diameter, followed by FE model simulation. The permutation approach employed for the EXP dataset is replicated. The NUM dataset comprises all data derived from the numerical signals. In contrast, the HYB dataset amalgamates the EXP and NUM datasets. For diameters present in both the experimental and FE model datasets meant for training, only experimental data is incorporated into the HYB dataset. These datasets subsequently inform the training phase of a 1D CNN algorithm to assess damage size. Performance is gauged using validation and test subsets sourced from the EXP dataset. The distinguishing factor between these datasets is the origin of training data; the HYB dataset integrates both experimental readings and numerical data extracted from the Bayesian stochastic FE model.

5.3.4 1D CNN structure definition

The proposed 1D CNN consists of three sequence blocks of convolution-pooling-ReLU layers, followed by one fully connected layer, as presented in Tab. 14. The three blocks of

1D Convolutions + ReLU layers aim to extract features from the time series. The network has increasing filters along its depth, and the last layers comprise a sequence of flattened, fully connected, and regression layers. In this structure, the convolution part acts as a low-level feature extractor, while the fully connected part performs the regression based on such features. The last layer is a regression layer with a single neuron that predicts the diameter of the damage between 20 and 80 mm. The error is quantified using the mean squared error, the prediction, and the real value. The same structure is used in all training scenarios to evaluate the data impact on the classifier's performance.

The internal structure of the network is evaluated through random search. For the first layer, two variables of interest are considered: kernel size and the number of filters per convolution. As the first layer is responsible for extracting basic features from the raw signal, the second and third kernel sizes are kept constant to reduce the hyperparameters search space. However, the number of filters is increased with the depth of the network; thus, the second and third layers have 2x and 3x times the number of filters of the first layer, respectively.

At the training stage, five hundred neural networks are simulated with hyperparameters sampled within the range $\alpha \sim \mathcal{U}\big(0.1, 0.00001\big)$, kernel size $K \sim \mathcal{U}\big(3, 25\big)$, and the number of filters $N \sim \mathcal{U}\big(16, 32\big)$. It is worth noting that only odd numbers for the kernel size are evaluated to maintain symmetry around the convolution window. The adaptive moment estimation (Adam) training algorithm is used. The other training hyperparameters kept constant are presented in Tab. 12. Besides a maximum of 500 training epochs, early stop criteria are implemented to reduce overfitting. The validation data from the EXP dataset is used to evaluate the loss and RMSE (see Tab. 8) along the training, and it is interrupted if validation loss stops decreasing or starts to increase in two subsequent epochs. The neural network structures are implemented in the MATLAB deep learning toolbox. All training rounds are carried out on a PC featuring AMD Ryzen 9 5950X CPU @ 3.40 GHz, 16 GB RAM, and an Nvidia GeForce 2060 GPU card using the GPU.

The results and hyperparameters for the three best-performing networks are presented in Tab. 13 and are ordered based on the validation data error for the network trained with experimental data. The best networks exhibit similar errors in the training of the numerical network, rendering this criterion irrelevant to the selection process.

Figure 62 compares the networks of Tab. 13 during the experimental and hybrid data training stage. Networks 1 and 3 have learning rates in the same order of magnitude and

Table 12 – Fixed training parameters.

Training parameters	
Optimizer	Adam
β_1	0.9
β_2	0.999
ϵ	1E-8
L2 regularization factor	0.0001
Weight initializer	He
Total epochs	500
Mini-batch size	128

Source: Prepared by the author.

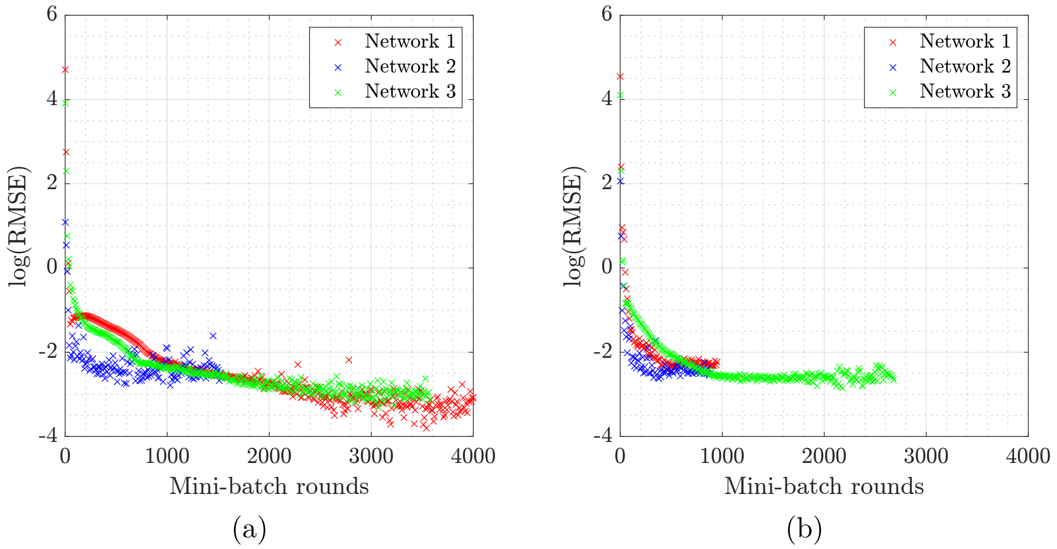
Table 13 – Parameters and results on validation data for the 3 best networks at hyperparameters tuning phase.

	Hyperparameters			**Error at validation data**		
Network	**Learning rate**	**Kernel size**	**Number of filters**	**EXP**	**NUM**	**HYB**
1	0.06720	19	31	0.0283	0.0817	0.0884
2	0.00360	3	24	0.0317	0.0819	0.0733
3	0.05640	19	20	0.0412	0.0780	0.0493

Source: Prepared by the author.

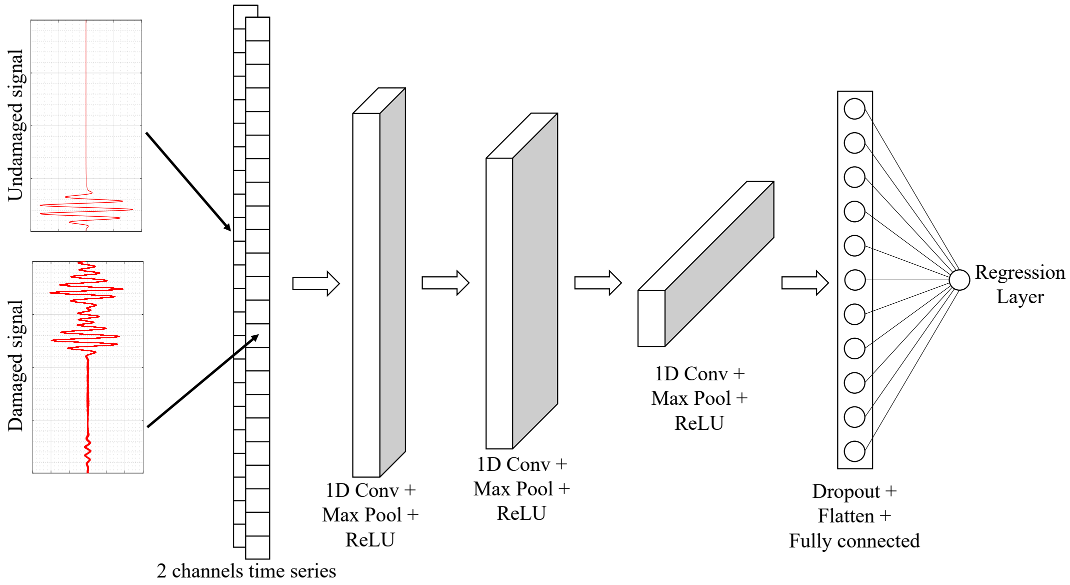
the same kernel size, but network 1 has more filters. A larger number of filters results in a network with more parameters to train. Network 1 exhibits a higher tendency for instability in the validation data after a certain point, indicating overfitting to the training data. Network 2 encounters a similar stability issue and undergoes early stopping. Conversely, when examining the curve for case number 3, a more stable behavior throughout the training process is observed, allowing for additional training rounds.

Consequently, the network architecture is selected based on a trade-off between the error in experimental and hybrid training, the tendency to overfitting, and the training time required for the subsequent stages. Figure 63 depicts the networks architecture and Tab. 14 presents the parameters for the selected network.



Source: Prepared by the author.

Figure 62 – Validation RMSE for the three best networks with: (a) EXP dataset; and (b) HYB dataset.



Source: Prepared by the author.

Figure 63 – 1D CNN architecture used for damage quantification.

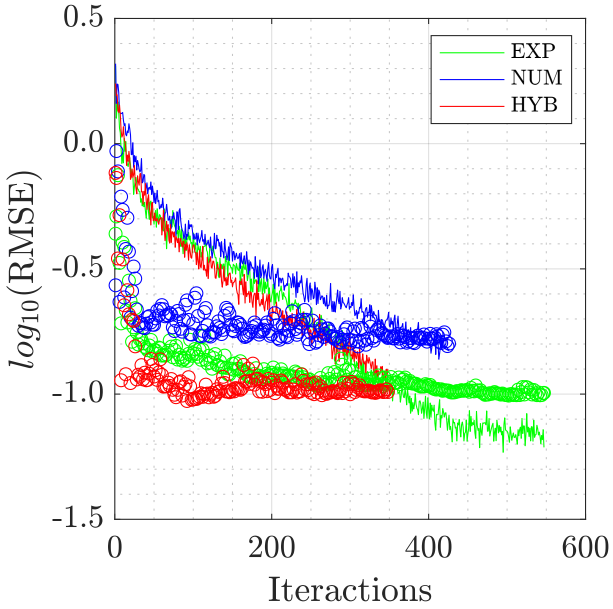
Table 14 – Network structure used for regression.

Layer	Type	Kernel	Stride
1	Input	-	-
2	1D convolution	19x20	1
3	Batch normalization	-	-
4	ReLU		-
5	Max pooling	-	2
6	1D convolution	10x40	1
7	Batch normalization	-	-
8	ReLU		-
9	Max pooling	-	2
10	1D convolution	10x60	1
11	Batch normalization	-	-
12	ReLU	5	-
13	Max pooling		2
14	Dropout	50%	-
15	Flatten		-
16	Fully connected		-

Source: Prepared by the author.

5.3.5 Damage quantification

The selected network is assessed using the EXP, NUM, and HYB datasets described in Section 5.3.3. The learning rate is initially set as 0.001 to minimize instability and subsequently fine-tuned throughout the process. An early stopping criterion based on the validation data error from the EXP dataset is employed. Training is stopped if the mean squared error decreases for two consecutive epochs. Figure 64 shows the training performance for each scenario.



Source: Prepared by the author.

Figure 64 – Comparison between training (—) and validation RMSE (∘) for the EXP, NUM, and HYB training strategies.

The algorithm trained with the EXP dataset reaches a training error approximately 20% lower than the validation error, indicating overfitting to the training data. The algorithms trained with the NUM and HYB datasets exhibit comparable validation-to-training error ratios, although the HYB-trained algorithm demonstrates lower absolute error values. Furthermore, the algorithm trained with the HYB dataset achieves a lower validation error faster than the other two, which combined with the fact that the validation error is similar in the EXP and HYB strategies, demonstrates the greater robustness of the algorithm trained with the inclusion of data from the Bayesian stochastic FE model.

Table 15 summarizes the average results, with the model trained using the EXP dataset as the reference for error comparison. When evaluated on the test data, the model trained with the HYB dataset exhibits an average RMSE that is 87.5% of the model trained with the EXP dataset, resulting in a 12.5% reduction in error. The HYB model shows a 167% higher error on the training data than the EXP model. However, this difference can be attributed to the overfitting of the EXP model on the training data. Overfitting may lead to the model fitting too closely to the training data, resulting in poorer performance when faced with new, unseen data. Overfitting metrics reveal that the EXP model has validation/training and test/training ratios of 5.74 and 9.54, respectively, while the HYB model exhibits ratios of 2.16 and 3.11 for the same metrics. These findings suggest that the algorithm trained with the HYB dataset possesses superior generalization capabilities.

Conversely, training with the NUM dataset has a considerably higher error, as shown in Tab. 15. As this algorithm is trained with only numerical data and validated with experimental data for the early stopping criteria throughout the training, the training process stops when it starts to overfit the numerical data. Although the stochastic FE model replicates most of the signal within the confidence interval, as seen in Figs. 61a to 61d, then the neural network starts to learn the particularities of the numerical signal and overfits after a certain training time. Additional tests on the NUM dataset have revealed that the proposed 1D CNN cannot effectively generalize when solely trained with numerical data in this experimental setup. One hypothesis is that, in this scenario, as the 1D CNNs are trained with the NUM dataset but validated with the EXP dataset throughout training, the training process may be terminated prematurely by the training algorithm. This leads to an underfitted model for both the numerical and experimental data, as shown by the high training RMSE. Consequently, this model is not evaluated in the subsequent analysis.

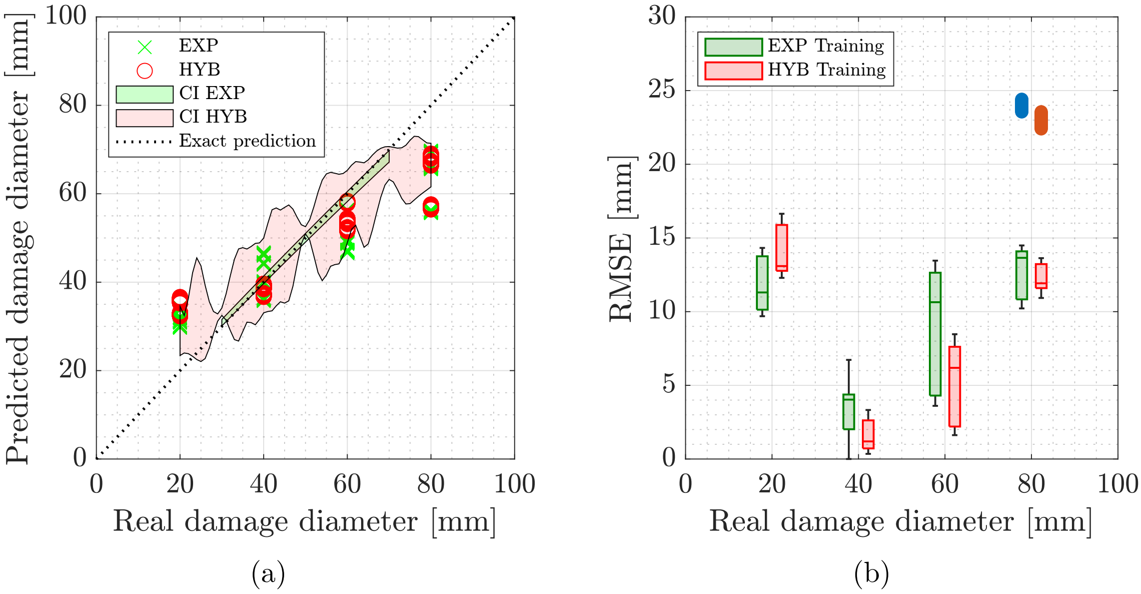
Table 15 – Summary of the quantification results.

Training strategy	**EXP**	**NUM**	**HYB**
Training RMSE (TR) [mm]	1.01	7.09	2.69
As EXP percentual	100%	705%	267%
Validation RMSE (VL) [mm]	5.77	9.46	5.83
As EXP percentual	100%	163%	101%
Test RMSE (TE) [mm]	9.59	12.3	8.38
As EXP percentual	100	128%	87.5%
Overfiting metrics			
VL/TR Ratio	5.74	1.33	2.16
TE/TR Ratio	9.54	1.73	3.11
TE/VL Ratio	1.66	1.30	1.44

Source: Prepared by the author.

Figure 65 compares the training strategies in terms of the classification of the experimental data. The confidence interval shown is derived from the 99% percentile of predictions on the training data. The neural network trained solely on experimental data exhibits a narrow confidence interval closely aligned with the optimal prediction region. However, this confidence interval does not accurately reflect reality due to its limited training data. All predictions for values in the test data lie outside the confidence interval, indicating the low reliability of this classifier.

In contrast, the neural network trained with the HYB dataset demonstrates a more reliable confidence interval, as shown in Fig. 65a. Note that the confidence interval is narrow around the training diameters of 30, 50, and 70 mm and widens as it deviates from these regions. The network's training data includes experimental realizations at these diameters, leading to narrower confidence intervals. In contrast, the stochastic model generates the potential signals for other diameters in the analysis range. The neural network predictions trained with hybrid data mostly fall within the confidence interval, except for some predictions at 20 and 80 mm. These damages occur at the extreme ends of the FE model's validity range and in the extrapolation region of the experimental data used during training. Figure 65b shows that the neural network trained with the HYB dataset exhibits smaller errors within the interpolation range of the model. The error increases in regions near the validity limit of the stochastic model, but the network trained with HYB dataset still shows less dispersion in its predictions.

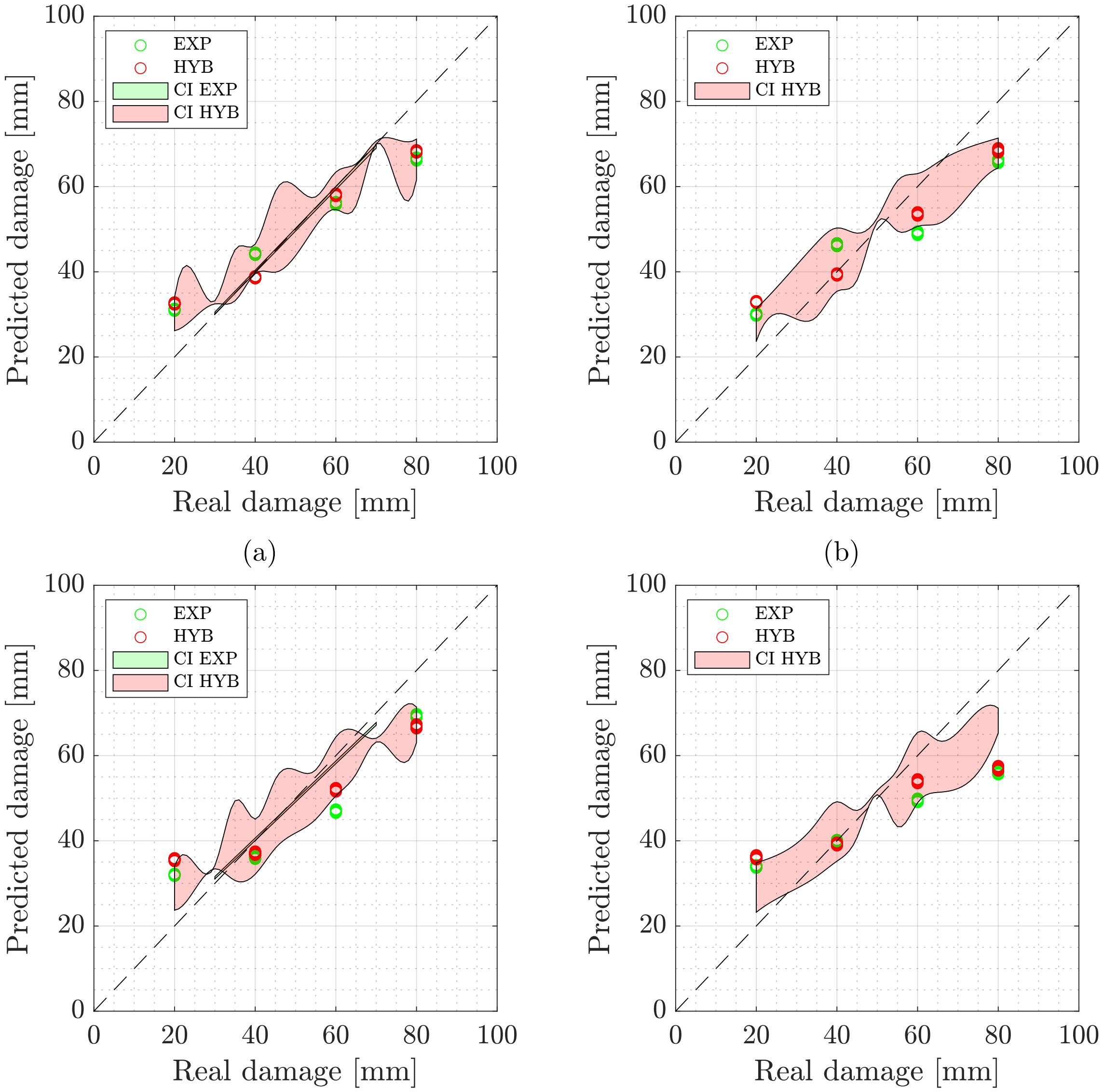


Source: Prepared by the author.

Figure 65 – 1D CNN predictions for experimental test data: (a) Real and predicted damages along with confidence intervals (CI); and (b) RMSE concerning real diameter.

5.3.6 Temperature influence

Figure 66 displays the 1D CNN-based classifier predictions for training with the EXP and HYB datasets, divided by temperature. Upon evaluating the predictions, it is noticeable that the classifier trained using the HYB dataset outperforms the one trained with the EXP dataset at all temperatures within the interpolation range (between 30 and 70 mm). Nevertheless, the classifier trained with the HYB dataset exhibits lower error than the classifier trained with the EXP dataset and can provide a similar confidence interval for multiple temperatures. Note that the model trained with the EXP dataset only has access to a single experimental condition for temperatures of 10°C and 60°C. Therefore, it lacks a confidence interval at these temperatures.



(a)

(b)

(c)

(d)

Source: Prepared by the author.

Figure 66 – 1D CNN performance in different temperatures: (a) 0° C; (b) 10° C; (c) 30° C; and (d) 60° C. CI represents the confidence intervals.

5.4 DISCUSSIONS

As depicted in Figs 65 and 66, the 1D CNN trained with the proposed Bayesian data-driven strategy exhibits lower error and can provide reliable confidence intervals. Observing the narrower confidence interval at damage diameters where the experimental signal is used at the training step is possible. At the same time, it widens at diameters trained only with numerical data.

As the algorithm trained with the EXP and HYB datasets has to learn from both sources, it is harder to overfit the training data when compared to the model trained with the EXP dataset. The latter presents a smaller error value on training data without a corresponding reduction in validation and test data. Despite resulting in a final validation error close to that observed in hybrid training, the ratio validation/training error is much higher for experimental training, indicating overfitting.

The region spanning from 30 to 70 mm corresponds to the interpolation segment of the training data, whereas 20 and 80 mm denotes extrapolation. Notably, the average error rises within the extrapolation regions. This can be attributed to both the inherent challenges associated with extrapolation and the proximity to the validity boundaries of the numerical model, which underwent validation between 20 and 80 mm.

Conversely, the network trained with the NUM dataset cannot achieve the same error level as the other two. It performs poorly in all scenarios due to the stopping criterion based on experimental data. Thus, it cannot train for sufficient time to adjust to the training data before the results diverge. For this network, the relaxation of the stopping criterion, allowing for more extended training, was evaluated. However, while the network improved performance on numerical data, it considerably worsened on experimental validation and testing data. Therefore, this training approach proved unfeasible with the given database, model, and training strategy.

In Fig. 66, the model exhibits superior performance at lower temperatures than higher ones, which can be attributed to two factors. Firstly, the effect of the simulated damage on Lamb wave propagation is highly temperature-dependent due to the viscoelastic nature of the material used to simulate the damage, and its influence diminishes at higher temperatures. This reduction is mainly due to reduced damping, as shown in Figs. 61a to 61d. Secondly, the temperature range between the first three experimental datasets spans 30 degrees, while the difference between the 30 and 60-degree datasets is also 30 degrees. This requires the classifier to work across a wider range at higher temperatures in addition to the reduced damage effect.

In situations with limited experimental data, 1D CNNs trained solely on experimental data cannot provide a confidence interval, as was the case at 10°C and 60°C. Conversely, the 1D CNN trained with the HYB dataset incorporates multiple potential signals generated by the stochastic model within its confidence interval, enabling the machine learning algorithm to establish its confidence intervals based on numerical data. The confidence intervals allow for greater confidence in the neural network's results, mitigate the black box nature typically associated with this class of algorithms, and allow one to explore the algorithm predictions across the proposed training scenarios.

5.5 CONCLUSIONS

The proposed Bayesian data-driven framework identifies and integrates relevant FE model input parameters from experimental data while accounting for their inherent uncertainties. As a result, the stochastic FE model completes and expands the existing dataset by providing simulated samples where data might be sparse. Combined with the generalization capabilities of 1D CNNs, this approach results in a model that presents accelerated convergence, minimized validation and test errors, and superior resilience to temperature variations.

The samples produced by the stochastic FE model can be merged during the training of machine learning algorithms, thereby enhancing these algorithms accuracy in damage quantification. As evidenced in the tests, the 1D CNN trained using this model-enhanced data achieved a 12.5% reduction in test data error compared to an algorithm trained using only experimental data.

Moreover, this expanded dataset paves the way for establishing a confidence interval for model predictions. This addition significantly boosts the reliability of results from the neural network, a crucial feature given the network's typical "black-box" nature.

In scenarios characterized by diverse damage types and locations, the challenges faced by algorithms trained only on experimental data intensify. The proposed framework is flexible, and can be applied to such situations, given that there is a FE model capable of simulating such damages. The approach reduces the difficulty and amount of required experiments in a multidimensional landscape, offering an alternative solution in situations where exhaustive factor combination testing is prohibitive.

6 FINAL REMARKS

There is still a long journey until Lamb waves become a widespread technique for industrial applications. However, small steps toward better algorithms and reliable data bring us closer to this objective. The work contained in this thesis explores new contributions to the application of 1D CNNs as data processing algorithms for Lamb wave signals, serving roles in detection, localization, and quantification. Additionally, a stochastic updating strategy based on Bayesian inference for updating FE models is presented. The data generated by these stochastic FE models can be utilized to train machine learning algorithms, complementing experimental data.

Thus, Section 6.1 presents a summary of the conclusions presented throughout this study. Additionally, Section 6.2 proposes paths for future inquiry, and Section 6.3 presents the research papers published during the development of this Thesis.

6.1 SUMMARY

- Results presented in Chapter 3 indicate that 1D CNNs are considerably better than their MLP counterparts in detecting and localizing damage. In a dataset with 196 damage positions collected with an automated version of the LaWaDe system from GRAVi UFMG, these networks consistently outperformed the MLP, showcasing its effectiveness as a feature extractor, classifier, and regressor for Lamb wave signals. The main conclusion can be summarized on two fronts:
  - These networks exhibit superior performance by utilizing the entire time series, avoiding subsampling, transformations, or loss of information. Consequently, they can directly map high-dimensional data from the complete Lamb wave signal to the desired outputs.
  - The 1D CNNs outperform MLP networks in both global and local damage localization strategies. For global localization, 1D CNNs achieve nearly 100% accuracy on test data when trained with 25 and 16 damage samples per 200 ×

200 mm quadrant. They maintain a median accuracy above 85% even with a 64% reduction in data, using only 9 samples per quadrant. In local analysis, 1D CNNs excel further, demonstrating lower mean error in predicted coordinates even with less training data compared to MLPs trained with maximal data.

  - A significant drawback of 1D CNNs is their complexity and computational cost in configuration and training. In contrast to MLPs, which have only three architectural hyperparameters—number of layers, number of neurons, and dropout probability—1D CNNs present a multitude of possible architectures with numerous hyperparameters for each layer. This complexity significantly complicates the fine-tuning process of these networks, making it both challenging and time-intensive.

- The influence of environmental conditions, notably temperature, on Lamb wave propagation is a well-documented issue in the literature. Furthermore, it is essential to consider experimental uncertainty in the development of a reliable SHM classification algorithm. Although the integration of these effects into FE models through Bayesian inference to enhance their ability to generate realistic data is a promising approach, the literature review shows a lack of research in this area. The primary challenges involve the complexity of Lamb waves, measurement uncertainty, and the computational demands of simulating Lamb wave models within a Bayesian framework. Chapter 4 presents results demonstrating the feasibility of using Bayesian inference in Lamb wave FE models to incorporate experimental uncertainty, thereby enhancing model reliability and its utility in generating data for training machine learning models. Additionally, a surrogate modeling strategy employing an artificial neural network was developed to mitigate computational costs. The results indicate that a neural network-based surrogate model can effectively replace an FE model for Lamb wave simulations, even within a Bayesian inference framework, with minimal impact on result quality. This surrogate modeling approach has potential applications in various other domains, such as vibration and buckling problems.

- Results in Chapter 5 indicate the Bayesian data-driven framework excels in incorporating key FE model parameters from experimental data, considering their uncertainties. This enhances the stochastic FE model, allowing it to supplement sparse datasets with simulated samples. When combined with 1D CNNs, this methodology leads to faster convergence, lower validation and test errors, and better resistance to temperature changes. Using this framework's output in machine learning algorithm training improves damage quantification accuracy. The tests show that algorithms trained with this enhanced data reduce test data errors by 12.5% compared to those trained solely on experimental data. Furthermore, this enriched dataset enables the establishment of confidence intervals for model predictions, increasing the neural network's reliability. This is especially useful given the network's ‘‘black-box'' nature. The proposed framework is also versatile and can be adapted to complex situations with varied damage types and locations, reducing the need for extensive experiments in multidimensional settings and offering a practical solution where comprehensive factor testing is impractical.

6.2 PERSPECTIVES OF FUTURE WORK

Some of the proposals for future work have already been introduced throughout the concluding remarks of each Chapter (please, see Sections 3.5, 4.5, and 5.5). Some other specific suggestions for future research that have emerged in the progress of this work are outlined next:

6.2.1 Model-based transfer learning using 1D CNNs

The work done in this research can be expanded with a transfer learning methodology, utilizing Lamb wave signals from the stochastic model to real structures. This could be achieved through domain adaptation or by pre-training a model with the stochastic model and then fine-tuning it using experimental data. In the context of 1D CNNs, this concept can be operationalized by modifying the first and last layers of the network.

Furthermore, this methodology could also be assessed using experimental data from the GRAVi dataset. It is important to note that the 4x4 grid array illustrated in Fig. 22 can be segmented into four distinct 3x3 arrays, analogous to the configuration used in Chapter 3. Various strategies could be explored, including the symmetry of the sensor mesh, mirroring the structure, and utilizing a network trained on the four upper-right quadrants to detect damage in other quadrants, particularly the lower-left quadrant, due to the plate's symmetry.

The concept of a global-local strategy could be expanded. Initially, this involves determining the location of the quadrant among the nine quadrants of the plate. Subsequently, a specialized 1D CNN, which only receives inputs from the four sensors demarcating a quadrant, could be employed to locate damage within all quadrants using this singular network.

6.2.2 Expand the stochastic model updating strategy for damage localization

In Chapter 4, a stochastic FE model of the CONCEPT was developed, focusing solely on varying the damage size. A promising direction for future research involves extending this model to support the GRAVi setup, integrating multiple damage sizes and positions. This enhancement aims to assess the capabilities of the stochastic updating framework and the processing efficacy of 1D CNNs with more complex datasets.

Due to computational limitations in the current methodology, it was not feasible to apply the proposed approach to larger structures, such as the GRAVi UFMG plate, using multiple sensors and actuators. The GRAVi plate, being twice the size of the CONCEPT dataset plate, necessitates a mesh four times larger. This scaling significantly increases the computational demands for Lamb wave simulations, necessitating either a high-performance workstation or a coordinated effort of multiple computers operating in parallel. In this context, the deployment of a surrogate model to reduce computational time becomes critically important.

An additional area of research that presents significant challenges involves models with damage in multiple positions, especially concerning Lamb waves. The high-speed propagation of these waves, on the order of kilometers per second, means that even millimetric discrepancies in damage positioning—between the actual damage, its recorded experimental location, and its representation in the numerical model—can lead to substantial phase differences in the wave packet. Addressing this precision issue is imperative for the successful application and validation of the model in real-world scenarios.

6.2.3 Experimental and numerical analysis of real damage using a stochastic model

The Bayesian data-driven framework for SHM presented in Chapter 5 demonstrated that, in some cases, the 1D CNNs have difficulty in generalizing based solely on numerical data, and require a certain amount of experimental data during the training phase to perform well in unseen experimental scenarios. This limitation is particularly significant in experimental scenarios featuring multiple damage types and/or when experimental data is expensive to collect. Therefore, an opportunity for research lies in the inclusion of real damage and in the elaboration of numerical models with different levels of fidelity.

This approach would aid in evaluating the generalization limitations of the 1D CNNs in real damage scenarios and provide guidelines indicating which types of damage should be collected experimentally and which cases can be simulated.

To optimize research efforts, the focus should initially be on damage types in composite structures that are relatively simple to replicate both experimentally and numerically, like holes. Additionally, prioritizing damages with substantial existing experimental data, such as delaminations, is recommended due to their critical importance.

6.3 CONTRIBUTIONS TO THE LITERATURE

This Thesis contributes to the literature by demonstrating the effectiveness of 1D CNNs in detecting structural damage using raw Lamb wave signals with minimal preprocessing. This finding highlights the potential of 1D CNNs in Lamb wave signal processing and opens new avenues for research of their application in SHM. Additionaly, a key contribution is the development of a framework for updating Lamb wave FE models to include temperature variations and measurement uncertainty, a first work in this field in the literature. The proposed approach enhances the accuracy of computational modeling in structural analysis. The Thesis also introduces a neural network-based surrogate model to simulate Lamb wave signals that increases the speed of Bayesian inference rounds. This model is currently being tested in carbon structures' buckling analysis and bridge vibration models, showing favorable results. Finally, the work outlines a method for training a 1D CNN with data from both experimental setups and a stochastic FE model. This approach effectively integrates theoretical and practical data, improving accuracy in structural damage detection. Some of the results of this Thesis were published in the following journal articles:

- **Leonardo de Paula S. Ferreira**, Rafael de O. Teloli, Samuel da Silva, Eloi Figueiredo, Ionut D. Moldovan, Nuno Maia, Carlos Alberto Cimini Jr., *Bayesian calibration for Lamb wave propagation on a composite plate using a machine learning surrogate model*, Mechanical Systems and Signal Processing, 2024.
- **Leonardo de Paula S. Ferreira**, Rafael de O. Teloli, Samuel da Silva, Eloi Figueiredo, Nuno Maia, Carlos Alberto Cimini Jr., *Bayesian data-driven framework for structural health monitoring of composite structures under limited experimental data*, Structural Health Monitoring, 2024.

There is also a manuscript under review related to this work:

- **Leonardo de Paula S. Ferreira**, Lázaro Valentim Donadon, Carlos Alberto Cimini Jr., *Damage localization on composite structures usingLamb waves and 1D CNNs*, Smart Materials and Structures, 2024.

The following Conferences were attended in the period:

- **Leonardo Ferreira**, Rafael de O. Teloli, Samuel da Silva, Eloi Figureiredo, Ionut D. Moldovan,Nuno Maia, Carlos Alberto Cimini Jr., *Recalage stochastique d'un modèle de propagation des ondes ultrasonores àl'aide d'un modèle de substitution à apprentissage automatique.* Journeé Nationale sur les Composites (JNC 23), Besançon, France, 2023.

- **Leonardo Ferreira**, Rafael Teloli, Samuel Da Silva, Eloi Figueiredo, Dragos Moldovan, Nuno Maia, Carlos Cimini Jr., *Stochastic Digital Twin of a Composite Plate for Predicting Lamb Wave Propagation.* Survailance, Vibrations, Shocks and Noise (SURVISHNO 23), Toulouse, France, 2023.

- **Leonardo de Paula S. Ferreira**, Marcos Omori Yano, Samuel da Silva, Carlos Alberto Cimini Júnior, *Damage quantification on composite structures using neural networks and hybrid data.* International Council of the Aeronautical Sciences (ICAS), Stockholm, Sweden, 2022.

- Marcus Omori Yano, **Leonardo de Paula S. Ferreira**, Samuel da Silva, Carlos Cimini, Eloi Figueiredo, *Gaussian process regression using hybrid data for damage quantification*, 6th Brazilian Conference on Composite Materials (BCCM), Tiradentes, Brasil, 2022

- **Leonardo de Paula S. Ferreira**, Lazaro Valentim Donadon, Paulo Henriques Iscold Andrade de Oliveira, *Ground vibration test using acoustic excitation: Application on a composite wing*, International Council of the Aeronautical Sciences (ICAS), Belo Horizonte, Brazil, 2018.

REFERENCES

ABDEL-HAMID, O.; MOHAMED, A.-r.; JIANG, H.; PENN, G. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In: **2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2012. p. 4277--4280.

ABDELJABER, O.; AVCI, O.; KIRANYAZ, S.; GABBOUJ, M.; INMAN, D. J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. **Journal of Sound and Vibration**, v. 388, p. 154 ,Äì 170, 2017. Cited by: 802. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84997079451&doi=10.1016%2fj.jsv.2016.10.043&partnerID=40&md5=675a85438b3524a26863e23902751bcd.

ACHENBACH, J. Quantitative nondestructive evaluation. **International Journal of Solids and Structures**, v. 37, n. 1, p. 13--27, 2000. ISSN 0020-7683. Available from: https://www.sciencedirect.com/science/article/pii/S0020768399000748.

APALOWO, R. K.; CHRONOPOULOS, D. A wave-based numerical scheme for damage detection and identification in two-dimensional composite structures. **Composite Structures**, Elsevier, v. 214, p. 164--182, 4 2019. ISSN 0263-8223.

AVCI, O.; ABDELJABER, O.; KIRANYAZ, S.; INMAN, D. Structural damage detection in real time: Implementation of 1d convolutional neural networks for shm applications. In: NIEZRECKI, C. (Ed.). [S.l.]: Springer International Publishing, 2017. p. 49--54. ISBN 978-3-319-54109-9.

AZUARA, G.; RUIZ, M.; BARRERA, E. Damage localization in composite plates using wavelet transform and 2-d convolutional neural networks. **Sensors**, v. 21, n. 17, 2021. ISSN 1424-8220. Available from: https://www.mdpi.com/1424-8220/21/17/5825.

BAISTHAKUR, S.; CHAKRABORTY, A. Modified hamiltonian monte carlo-based bayesian finite element model updating of steel truss bridge. **Structural Control and Health Monitoring**, v. 27, n. 8, p. e2556, 2020. E2556 stc.2556. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2556.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York, NY: Springer, 2006.

BITTER, R.; MOHIUDDIN, T.; NAWROCKI, M. **LabVIEW: Advanced programming techniques**. [S.l.]: Crc Press, 2006.

BOLLER, C.; TOMLINSON, G. R.; STASZEWSKI, W. J. (Ed.). **Health monitoring of aerospace structures : smart sensor technologies and signal processing**. Chichester, West Sussex, England ; Hoboken, NJ: J. Wiley, 2004. 266 p. Available from: https://onlinelibrary.wiley.com/doi/book/10.1002/0470092866.

BUBECK, S.; CHANDRASEKARAN, V.; ELDAN, R.; GEHRKE, J.; HORVITZ, E.; KAMAR, E.; LEE, P.; LEE, Y. T.; LI, Y.; LUNDBERG, S.; NORI, H.; PALANGI, H.; RIBEIRO, M. T.; ZHANG, Y. Sparks of artificial general intelligence: Early experiments with gpt-4. Mar 2023. ArXiv preprint arXiv:2303.12712. Available from: https://doi.org/10.48550/arXiv.2303.12712.

BUD, M. A.; MOLDOVAN, I.; RADU, L.; NEDELCU, M.; FIGUEIREDO, E. Reliability of probabilistic numerical data for training machine learning algorithms to detect damage in bridges. **Structural Control and Health Monitoring**, v. 29, n. 7, p. e2950, 2022. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2950.

CARRINO, S.; MAFFEZZOLI, A.; SCARSELLI, G. Active shm for composite pipes using piezoelectric sensors. **Materials Today: Proceedings**, v. 34, p. 1--9, 2021. ISSN 2214-7853. 12th International Conference on Composite Science and Technology. Available from: https://www.sciencedirect.com/science/article/pii/S2214785319340593.

CHEEKE, J. D. N. **Fundamentals and Applications of Ultrasonic Waves**. 2. ed. [S.l.]: CRC Press, 2017. 504 p. ISBN 978-1138077201.

CHIB, S.; GREENBERG, E. Understanding the metropolis-hastings algorithm. **The American Statistician**, Taylor & Francis, Ltd., v. 49, n. 4, p. 327--335, nov. 1995. Available from: https://doi.org/10.2307/2684568.

COMPOSITE materials handbook. [S.l.]: SAE International on behalf of CMH-17, a division of Wichita State University, 2017.

COURANT, R.; FRIEDRICHS, K.; LEWY, H. On the partial difference equations of mathematical physics. **IBM Journal of Research and Development**, IBM Corp., v. 11, p. 215--234, 1967. ISSN 00188646.

CRISTIANI, D.; FALCETELLI, F.; YUE, N.; SBARUFATTI, C.; Di Sante, R.; ZAROUCHAS, D.; GIGLIO, M. Strain-based delamination prediction in fatigue loaded cfrp coupon specimens by deep learning and static loading data. **Composites Part B: Engineering**, v. 241, p. 110020, 2022. ISSN 1359-8368. Available from: https://www.sciencedirect.com/science/article/pii/S1359836822003985.

CROXFORD, A. J.; MOLL, J.; WILCOX, P. D.; MICHAELS, J. E. Efficient temperature compensation strategies for guided wave structural health monitoring. **Ultrasonics**, v. 50, n. 4, p. 517--528, 2010. ISSN 0041-624X. Available from: https://www.sciencedirect.com/science/article/pii/S0041624X09001826.

CROXFORD, A. J.; WILCOX, P. D.; KONSTANTINIDIS, G.; DRINKWATER, B. W. Strategies for overcoming the effect of temperature on guided wave structural health monitoring. In: KUNDU, T. (Ed.). **Health Monitoring of Structural and Biological Systems 2007**. SPIE, 2007. v. 6532, p. 65321T. Available from: https://doi.org/10.1117/12.719435.

De Fenza, A.; SORRENTINO, A.; VITIELLO, P. Application of artificial neural networks and probability ellipse methods for damage detection using lamb waves. **Composite Structures**, v. 133, p. 390--403, 2015. ISSN 0263-8223. Available from: https://www.sciencedirect.com/science/article/pii/S0263822315006352.

De LUCA, A.; PERFETTO, D.; De Fenza, A.; PETRONE, G.; CAPUTO, F. Guided wave shm system for damage detection in complex composite structure. **Theoretical and Applied Fracture Mechanics**, v. 105, p. 102408, 2020. ISSN 0167-8442. Available from: https://www.sciencedirect.com/science/article/pii/S0167844219302460.

DENG, P.; SAITO, O.; OKABE, Y.; SOEJIMA, H. Simplified modeling method of impact damage for numerical simulation of lamb wave propagation in quasi-isotropic composite structures. **Composite Structures**, v. 243, p. 112150, 2020. ISSN 0263-8223. Available from: https://www.sciencedirect.com/science/article/pii/S0263822319336992.

DONADON, L. V.; FERREIRA, L. d. P. S.; DUARTE, H. V. Recent development in the structural health monitoring in the gravi group in ufmg. In: **Proceedings of the 23rd ABCM International Congress of Mechanical Engineering**. Rio de Janeiro, RJ, Brazil: [s.n.], 2015.

DONADON, L. V.; FERREIRA, L. d. P. S.; DUARTE, H. V.; MEDEIROS, E. B. Simple attempt to compensate temperature variation in structural health monitoring system using lamb wave inspection. In: **Proceedings of the 23rd ABCM International Congress of Mechanical Engineering**. Rio de Janeiro, RJ, Brazil: [s.n.], 2015.

DWORAKOWSKI, Z.; AMBROZINSKI, L.; PACKO, P.; DRAGAN, K.; STEPINSKI, T. Application of artificial neural networks for compounding multiple damage indices in lamb-wave-based damage detection. **Structural Control and Health Monitoring**, v. 22, n. 1, p. 50--61, 2015. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.1659.

ECKELS, J. D.; JACOBSON, E. M.; CUMMINGS, I. T.; FERNANDEZ, I. F.; HO, K.; DERVILIS, N.; FLYNN, E. B.; WACHTOR, A. J. Predicting local material thickness from steady-state ultrasonic wavefield measurements using a convolutional neural network. **Ultrasonics**, v. 123, p. 106661, 2022. ISSN 0041-624X. Available from: https://www.sciencedirect.com/science/article/pii/S0041624X2100278X.

EREIZ, S.; DUVNJAK, I.; JIMENEZ-ALONSO, J. F. Review of finite element model updating methods for structural applications. **Structures**, Elsevier Ltd, v. 41, p. 684--723, 7 2022. ISSN 23520124.

FAKIH, M. A.; CHIACHIO, M.; CHIACHIO, J.; MUSTAPHA, S. A Bayesian approach for damage assessment in welded structures using Lamb-wave surrogate models and minimal sensing. **NDT and E International**, Elsevier Ltd, v. 128, 6 2022. ISSN 09638695.

FARRAR, C. R.; DOEBLING, S. W.; NIX, D. A. Vibration-based structural damage identification. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 359, n. 1778, p. 131--149, 2001.

FENG, B.; RIBEIRO, A. L.; RAMOS, H. G. A new method to detect delamination in composites using chirp-excited lamb wave and wavelet analysis. **NDT and E International**, v. 100, p. 64--73, 2018. ISSN 0963-8695. Available from: https://www.sciencedirect.com/science/article/pii/S0963869518301506.

FRANCA, A. **Deteccao e localizacao de danos em materiais compositos aplicado em aeronaves utilizando redes neurais artificiais**. Thesis (Ph. D.) --- Universidade Estadual Paulista (Unesp), 2014.

FUKUSHIMA, K.; MIYAKE, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: AMARI, S.-i.; ARBIB, M. A. (Ed.). **Competition and Cooperation in Neural Nets**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982. p. 267--285. ISBN 978-3-642-46466-9.

GALLINA, A.; AMBROZINSKI, L.; PACKO, P.; PIECZONKA, L.; UHL, T.; STASZEWSKI, W. J. Bayesian parameter identification of orthotropic composite materials using lamb waves dispersion curves measurement. **Journal of Vibration and Control**, v. 23, p. 2656--2671, 2017. Available from: https://doi.org/10.1177/1077546315619264.

GIURGIUTIU, V. Preface. In: GIURGIUTIU, V. (Ed.). **Structural Health Monitoring with Piezoelectric Wafer Active Sensors (Second Edition)**. Second edition. Oxford: Academic Press, 2014. p. xi--xii. ISBN 978-0-12-418691-0. Available from: https://www.sciencedirect.com/science/article/pii/B9780124186910000204.

GONZALEZ-JIMENEZ, A.; LOMAZZI, L.; JUNGES, R.; GIGLIO, M.; MANES, A.; CADINI, F. Enhancing lamb wave-based damage diagnosis in composite materials using a pseudo-damage boosted convolutional neural network approach. **Structural Health Monitoring**, v. 0, n. 0, p. 14759217231189972, 2023. Available from: https://doi.org/10.1177/14759217231189972.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. http://www.deeplearningbook.org.

GORISSEN, D.; COUCKUYT, I.; DEMEESTER, P.; DHAENE, T.; CROMBECQ, K. A surrogate modeling and adaptive sampling toolbox for computer based design. **Journal of Machine Learning Research**, v. 11, p. 2051 ,Äì 2055, 2010. Cited by: 460. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-77956916106&partnerID=40&md5=ec1279fd372dd58d20ca4395758a19ea.

GRESIL, M.; GIURGIUTIU, V.; SHEN, Y.; PODDAR, B. Guidelines for using the finite element method for modeling of guided Lamb wave propagation in shm processes. **e-Journal of Nondestructive Testing (eJNDT) 1435-4934**, 2013. Available from: https://www.ndt.net/search/docs.php3?id=14011.

GUO, X.; LI, W.; IORIO, F. Convolutional neural networks for steady flow approximation. In: . [s.n.], 2016. v. 13-17-August-2016, p. 481 ,Äì 490. Cited by: 443. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84984994117&doi=10.1145%2f2939672.2939738&partnerID=40&md5=e800127c0e93d79b2e8931c1eb2acbcc.

HA, N.; LEE, H.-S.; LEE, S. Development of a wireless corrosion detection system for steel-framed structures using pulsed eddy currents. **Sensors**, v. 21, n. 24, 2021. ISSN 1424-8220. Available from: https://www.mdpi.com/1424-8220/21/24/8199.

HADI, M. U.; TASHI qasem al; QURESHI, R.; SHAH, A.; MUNEER amgad; IRFAN, M.; ZAFAR, A.; SHAIKH, M. B.; AKHTAR, N.; WU, J.; MIRJALILI, S. Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. 9 2023. Available from: https://www.techrxiv.org/articles/preprint/A_Survey_on_Large_Language_Models_Applications_Challenges_Limitations_and_Practical_Usage/23589741.

HAGHIGHAT, E.; RAISSI, M.; MOURE, A.; GOMEZ, H.; JUANES, R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. **Computer Methods in Applied Mechanics and Engineering**, v. 379, 2021. Cited by: 248. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102291184&doi=10.1016%2fj.cma.2021.113741&partnerID=40&md5=6a48001a0875e1ebcd4af1f2b9806439.

HAN, S. **Finite Element Analysis of Lamb Waves Acting within a Thin Aluminum Plate**. Dissertação (Master's thesis) --- Air Force Institute of Technology, Wright-Patterson AFB, Department of Aeronautics and Astronautics, set. 2007. Master's thesis.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. **arXiv preprint arXiv:1512.03385**, 2015. Tech report, Computer Vision and Pattern Recognition (cs.CV). Available from: https://doi.org/10.48550/arXiv.1512.03385.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. **arXiv preprint arXiv:1502.01852**, abs/1502.01852, 2015. Available from: https://doi.org/10.48550/arXiv.1502.01852.

HORA, P.; ČERVENÁ, O. Determination of lamb wave dispersion curves by means of fourier transform. **Applied and Computational Mechanics**, v. 6, n. 1, 2012.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Netw.**, Elsevier Science Ltd., GBR, v. 2, n. 5, p. 359,Äì366, jul 1989. ISSN 0893-6080.

HU, N.; LIU, Y.; LI, Y.; PENG, X.; YAN, B. Optimal excitation frequency of lamb waves for delamination detection in cfrp laminates. **Journal of Composite Materials**, v. 44, n. 13, p. 1643--1663, 2010. Available from: https://doi.org/10.1177/0021998309353965.

IHN, J.-B.; CHANG, F.-K. Pitch-catch active sensing methods in structural health monitoring for aircraft structures. **Structural Health Monitoring**, v. 7, n. 1, p. 5--19, 2008. Available from: https://doi.org/10.1177/1475921707081979.

INCE, T.; KIRANYAZ, S.; EREN, L.; ASKAR, M.; GABBOUJ, M. Real-time motor fault detection by 1-d convolutional neural networks. **IEEE Transactions on Industrial Electronics**, v. 63, n. 11, p. 7067--7075, 2016.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **CoRR**, abs/1502.03167, 2015. Available from: http://arxiv.org/abs/1502.03167.

ISA, D.; RAJKUMAR, R. Pipeline defect prediction using support vector machines. **Applied Artificial Intelligence**, v. 23, n. 8, p. 758 ,Äì 771, 2009. Cited by: 29. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-70350602497&doi=10.1080%2f08839510903210589&partnerID=40&md5=2ebc437f0dad5229cfe2225a032449e7.

JIN, S. S.; JU, H.; JUNG, H. J. Adaptive Markov chain Monte Carlo algorithms for Bayesian inference: recent advances and comparative study. **https://doi.org/10.1080/15732479.2019.1628077**, Taylor & Francis, v. 15, p. 1548--1565, 11 2019. ISSN 17448980. Available from: https://www.tandfonline.com/doi/abs/10.1080/15732479.2019.1628077.

KE, Y.-T.; CHENG, C.-C.; LIN, Y.-C.; HUANG, C.-L.; HSU, K.-T. Quantitative assessment of bonding between steel plate and reinforced concrete structure using dispersive characteristics of lamb waves. **NDT and E International**, v. 102, p. 311--321, 2019. ISSN 0963-8695. Available from: https://www.sciencedirect.com/science/article/pii/S0963869518303621.

KHAZAEE, M.; DERIAN, P.; MOURAUD, A. A comprehensive study on structural health monitoring (shm) of wind turbine blades by instrumenting tower using machine learning methods. **Renewable Energy**, v. 199, p. 1568--1579, 2022. ISSN 0960-1481. Available from: https://www.sciencedirect.com/science/article/pii/S0960148122013829.

KIRANYAZ, S.; AVCI, O.; ABDELJABER, O.; INCE, T.; GABBOUJ, M.; INMAN, D. J. 1d convolutional neural networks and applications: A survey. **Mechanical Systems and Signal Processing**, v. 151, p. 107398, 2021. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327020307846.

KONSTANTINIDIS, G.; DRINKWATER, B. W.; WILCOX, P. D. The temperature stability of guided wave structural health monitoring systems. **Smart Materials and Structures**, v. 15, n. 4, p. 967, jun 2006. Available from: https://dx.doi.org/10.1088/0964-1726/15/4/010.

KONSTANTINIDIS, G.; WILCOX, P. D.; DRINKWATER, B. W. An investigation into the temperature stability of a guided wave structural health monitoring system using permanently attached sensors. **IEEE Sensors Journal**, v. 7, n. 5, p. 905--912, 2007.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGES, C.; BOTTOU, L.; WEINBERGER, K. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2012. v. 25. Available from: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGES, C.; BOTTOU, L.; WEINBERGER, K. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2012. v. 25. Available from: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

KUDVA, J. N.; MUNIR, N.; TAN, P. W. Damage detection in smart structures using neural networks and finite-element analyses. **Smart Materials and Structures**, v. 1, n. 2, p. 108, jun 1992. Available from: https://dx.doi.org/10.1088/0964-1726/1/2/002.

LEE, I. Y.; JANG, J.; PARK, Y.-B. Advanced structural health monitoring in carbon fiber-reinforced plastic using real-time self-sensing data and convolutional neural network architectures. **Materials & Design**, v. 224, p. 111348, 2022. ISSN 0264-1275. Available from: https://www.sciencedirect.com/science/article/pii/S0264127522009704.

LEE, J.-S.; PARK, G.; KIM, C.-G.; FARRAR, C. R. Use of relative baseline features of guided waves for in situ structural health monitoring. **Journal of Intelligent Material Systems and Structures**, SAGE Publications Ltd STM, v. 22, p. 175--189, 1 2011. ISSN 1045-389X. Doi: 10.1177/1045389X10395643. Available from: https://doi.org/10.1177/1045389X10395643.

LEVY, M. Fundamentals and Applications of Ultrasonic Waves . **Physics Today**, v. 56, n. 4, p. 67--68, 04 2003. ISSN 0031-9228. Available from: https://doi.org/10.1063/1.1580055.

LI, P.; PEI, Y.; LI, J. A comprehensive survey on design and application of autoencoder in deep learning. **Applied Soft Computing**, v. 138, p. 110176, 2023. ISSN 1568-4946. Available from: https://www.sciencedirect.com/science/article/pii/S1568494623001941.

LI, X.; GUAN, Y.; LAW, S.; ZHAO, W. Monitoring abnormal vibration and structural health conditions of an in-service structure from its shm data. **Journal of Sound and Vibration**, v. 537, p. 117185, 2022. ISSN 0022-460X. Available from: https://www.sciencedirect.com/science/article/pii/S0022460X22003819.

LIM, H.; MANUEL, L. Distribution-free polynomial chaos expansion surrogate models for efficient structural reliability analysis. **Reliability Engineering & System Safety**, v. 205, p. 107256, 2021. ISSN 0951-8320. Available from: https://www.sciencedirect.com/science/article/pii/S0951832020307560.

LITJENS, G.; KOOI, T.; BEJNORDI, B. E.; SETIO, A. A. A.; CIOMPI, F.; GHAFOORIAN, M.; van der Laak, J. A.; van Ginneken, B.; SANCHEZ, C. I. A survey on deep learning in medical image analysis. **Medical Image Analysis**, v. 42, p. 60--88, 2017. ISSN 1361-8415. Available from: https://www.sciencedirect.com/science/article/pii/S1361841517301135.

LIU, C.; HARLEY, J. B.; BERGES, M.; GREVE, D. W.; OPPENHEIM, I. J. Robust ultrasonic damage detection under complex environmental conditions using singular value decomposition. **Ultrasonics**, v. 58, p. 75--86, 2015. ISSN 0041-624X. Available from: https://www.sciencedirect.com/science/article/pii/S0041624X14003515.

LIU, H.; ZHANG, Y. Deep learning based crack damage detection technique for thin plate structures using guided lamb wave signals. **Smart Materials and Structures**, IOP Publishing, v. 29, p. 15032, 12 2019. Available from: https://dx.doi.org/10.1088/1361-665X/ab58d6.

LIU, H.; ZHANG, Y. Deep learning based crack damage detection technique for thin plate structures using guided lamb wave signals. **Smart Materials and Structures**, IOP Publishing, v. 29, n. 1, p. 015032, dec 2019. Available from: https://dx.doi.org/10.1088/1361-665X/ab58d6.

LIU, S.; DU, C.; MOU, J.; MARTUA, L.; ZHANG, J.; LEWIS, F. Diagnosis of structural cracks using wavelet transform and neural networks. **NDT and International**, v. 54, p. 9--18, 2013. ISSN 0963-8695. Available from: https://www.sciencedirect.com/science/article/pii/S0963869512001545.

LIU, Z.; LESSELIER, D.; SUDRET, B.; WIART, J. Surrogate modeling based on resampled polynomial chaos expansions. **Reliability Engineering & System Safety**, v. 202, p. 107008, 2020. ISSN 0951-8320. Available from: https://www.sciencedirect.com/science/article/pii/S0951832020305093.

LOMAZZI, L.; FABIANO, S.; PARZIALE, M.; GIGLIO, M.; CADINI, F. On the explainability of convolutional neural networks processing ultrasonic guided waves for damage diagnosis. **Mechanical Systems and Signal Processing**, v. 183, p. 109642, 2023. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327022007282.

LU, Y.; YE, L.; SU, Z.; ZHOU, L.; CHENG, L. Artificial neural network (ann)-based crack identification in aluminum plates with lamb wave signals. **Journal of Intelligent Material Systems and Structures**, v. 20, n. 1, p. 39--49, 2009. Available from: https://doi.org/10.1177/1045389X07088782.

LUCA, A. D.; PERFETTO, D.; FENZA, A. D.; PETRONE, G.; CAPUTO, F. Guided wave SHM system for damage detection in complex composite structure. **Theoretical and Applied Fracture Mechanics**, Elsevier B.V., v. 105, 2 2020. ISSN 01678442.

LYE, A.; CICIRELLO, A.; PATELLI, E. Sampling methods for solving Bayesian model updating problems: A tutorial. **Mechanical Systems and Signal Processing**, v. 159, p. 107760, 2021. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327021001552.

MACE, B. R.; MANCONI, E. Modelling wave propagation in two-dimensional structures using finite element analysis. **Journal of Sound and Vibration**, Academic Press, v. 318, p. 884--902, 12 2008. ISSN 0022-460X.

MACE, B. R.; MANCONI, E. Modelling wave propagation in two-dimensional structures using finite element analysis. **Journal of Sound and Vibration**, Academic Press, v. 318, p. 884--902, 12 2008. ISSN 0022-460X.

MAHADEVKAR, S. V.; KHEMANI, B.; PATIL, S.; KOTECHA, K.; VORA, D. R.; ABRAHAM, A.; GABRALLA, L. A. A review on machine learning styles in computer vision,Äîtechniques and future directions. **IEEE Access**, v. 10, p. 107293--107329, 2022.

MARELLI, S.; SUDRET, B. UQLab: A framework for uncertainty quantification in Matlab. In: **Vulnerability, Uncertainty, and Risk**. ASCE Reston, VA, 2014. p. 2554--2563. Available from: https://ascelibrary.org/doi/abs/10.1061/9780784413609.257.

MARELLI, S.; SUDRET, B. **UQLab user manual -- Polynomial chaos expansions**. [S.l.], 2018. Report # UQLab-V1.1-104.

MARIANI, S.; RENDU, Q.; URBANI, M.; SBARUFATTI, C. Causal dilated convolutional neural networks for automatic inspection of ultrasonic signals in non-destructive evaluation and structural health monitoring. **Mechanical Systems and Signal Processing**, v. 157, p. 107748, 2021. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327021001436.

MARWALA, T. Finite-element-model updating using particle-swarm optimization. In: _____. **Finite-element-model Updating Using Computional Intelligence Techniques: Applications to Structural Dynamics**. London: Springer London, 2010. cap. 4, p. 67--84. ISBN 978-1-84996-323-7. Available from: https://doi.org/10.1007/978-1-84996-323-7_4.

MARZANI, A.; SALAMONE, S. Numerical prediction and experimental verification of temperature effect on plate waves generated and received by piezoceramic sensors. **Mechanical Systems and Signal Processing**, v. 30, p. 204--217, 2012. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327011004547.

MEAD, D. J. A general theory of harmonic wave propagation in linear periodic systems with multiple coupling. **Journal of Sound and Vibration**, v. 27, p. 235--260, 1973.

MECHBAL, N.; URIBE, J. S.; REBILLAT, M. A probabilistic multi-class classifier for structural health monitoring. **Mechanical Systems and Signal Processing**, v. 60-61, p. 106--123, 2015. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327015000230.

MIGUEL, L. P.; TELOLI, R. O.; da Silva, S. Bayesian model identification through Harmonic Balance Method for hysteresis prediction in bolted joints. **Nonlinear Dynamics**, v. 107, n. 1, p. 77--98, 2022. ISSN 1573-269X.

MITRA, M.; GOPALAKRISHNAN, S. Guided wave based structural health monitoring: A review. **Smart Materials and Structures**, IOP Publishing, v. 25, n. 5, p. 053001, mar 2016. Available from: https://dx.doi.org/10.1088/0964-1726/25/5/053001.

MOSER, F.; JACOBS, L. J.; QU, J. Modeling elastic wave propagation in waveguides with the finite element method. **NDT and E International**, Elsevier Science Ltd, v. 32, p. 225--234, 1999. ISSN 09638695.

MUSTAPHA, S.; YE, L.; DONG, X.; ALAMDARI, M. M. Evaluation of barely visible indentation damage (bvid) in cf/ep sandwich composites using guided wave signals. **Mechanical Systems and Signal Processing**, v. 76-77, p. 497--517, 2016. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S088832701600025X.

NAGY, P. B.; SIMONETTI, F.; INSTANES, G. Corrosion and erosion monitoring in plates and pipes using constant group velocity Lamb wave inspection. **Ultrasonics**, v. 54, n. 7, p. 1832--1841, 2014. ISSN 0041-624X. Available from: https://www.sciencedirect.com/science/article/pii/S0041624X14000183.

National Transportation Safety Board. **Aviation Accident Database & Synopses**. 2023. Available from: https://www.ntsb.gov/Pages/AviationQueryV2.aspx.

NG, C.; VEIDT, M.; ROSE, L.; WANG, C. Analytical and finite element prediction of lamb wave scattering at delaminations in quasi-isotropic composite laminates. **Journal of Sound and Vibration**, v. 331, n. 22, p. 4870--4883, 2012. ISSN 0022-460X. Available from: https://www.sciencedirect.com/science/article/pii/S0022460X12004671.

PACKO, P.; BIELAK, T.; SPENCER, A. B.; STASZEWSKI, W. J.; UHL, T.; WORDEN, K. Lamb wave propagation modelling and simulation using parallel processing architecture and graphical cards. **Smart Materials and Structures**, IOP Publishing, v. 21, n. 7, p. 075001, may 2012. Available from: https://dx.doi.org/10.1088/0964-1726/21/7/075001.

PADIL, K. H.; BAKHARY, N.; HAO, H. The use of a non-probabilistic artificial neural network to consider uncertainties in vibration-based-damage detection. **Mechanical Systems and Signal Processing**, v. 83, p. 194--209, 2017. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327016301893.

PAIVA, J. M. F.; MAYER, S.; REZENDE, M. C. Comparison of tensile strength of different carbon fabric reinforced epoxy composites. **Materials Research**, ABM, ABC, ABPol, v. 9, n. Mat. Res., 2006 9(1), Jan 2006. ISSN 1516-1439. Available from: https://doi.org/10.1590/S1516-14392006000100016.

PAIXAO, J.; SILVA, S. da; FIGUEIREDO, E.; RADU, L.; PARK, G. Delamination area quantification in composite structures using gaussian process regression and auto-regressive models. **Journal of Vibration and Control**, v. 27, n. 23-24, p. 2778--2792, 2021. Available from: https://doi.org/10.1177/1077546320966183.

PANDEY, P.; RAI, A.; MITRA, M. Explainable 1-d convolutional neural network for damage detection using lamb wave. **Mechanical Systems and Signal Processing**, Academic Press, v. 164, p. 108220, 2 2022. ISSN 0888-3270.

PARK, G.; FARRAR, C. R.; SCALEA, F. L. di; COCCIA, S. Performance assessment and validation of piezoelectric active-sensors in structural health monitoring. **Smart Materials and Structures**, v. 15, n. 6, p. 1673, oct 2006. Available from: https://dx.doi.org/10.1088/0964-1726/15/6/020.

PERFETTO, D.; Sharif Khodaei, Z.; De Luca, A.; ALIABADI, M.; CAPUTO, F. Experiments and modelling of ultrasonic waves in composite plates under varying temperature. **Ultrasonics**, v. 126, p. 106820, 2022. ISSN 0041-624X. Available from: https://www.sciencedirect.com/science/article/pii/S0041624X22001263.

RAI, A.; MITRA, M. Lamb wave based damage detection in metallic plates using multi-headed 1-dimensional convolutional neural network. **Smart Materials and Structures**, IOP Publishing, v. 30, p. 035010, 2 2021. ISSN 0964-1726.

RAI, A.; MITRA, M. A transfer learning approach for damage diagnosis in composite laminated plate using lamb waves. **Smart Materials and Structures**, IOP Publishing, v. 31, n. 6, p. 065002, apr 2022. Available from: https://dx.doi.org/10.1088/1361-665X/ac66aa.

RAMALHO, G. M. F.; LOPES, A. M.; CARBAS, R. J. C.; SILVA, L. F. M. D. Identifying weak adhesion in single-lap joints using lamb wave data and artificial intelligence algorithms. **Applied Sciences**, v. 13, n. 4, 2023. ISSN 2076-3417. Available from: https://www.mdpi.com/2076-3417/13/4/2642.

RAQUETI, R. da S.; TELOLI, R. de O.; SILVA, S. da; BUSSETTA, P.; CUNHA, J. A. On the use of stochastic Bouc‚ÄìWen model for simulating viscoelastic internal variables from a finite element approximation of steady-rolling tire. **Journal of**

Vibration and Control, v. 29, p. 10775463221125038, 2022. Available from: https://doi.org/10.1177/10775463221125038.

RITTO, T. G.; SAMPAIO, R.; AGUIAR, R. R. Uncertain boundary condition bayesian identification from experimental data: A case study on a cantilever beam. **Mechanical Systems and Signal Processing**, Academic Press, v. 68-69, p. 176--188, 2 2016. ISSN 10961216.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533--536, 1986. Available from: https://doi.org/10.1038/323533a0.

SALMANPOUR, M.; KHODAEI, Z. S.; ALIABADI, M. Guided wave temperature correction methods in structural health monitoring. **Journal of Intelligent Material Systems and Structures**, v. 28, n. 5, p. 604--618, 2017. Available from: https://doi.org/10.1177/1045389X16651155.

SANTANA, G. A. **A Numerical and Experimental Study on Structural Health Monitoring of Large Composite Shell-like Structures**. Thesis (Ph.D. dissertation) --- Universidade Federal de Minas Gerais, 2019.

SARMADI, H.; KARAMODIN, A. A novel anomaly detection method based on adaptive mahalanobis-squared distance and one-class knn rule for structural health monitoring under environmental effects. **Mechanical Systems and Signal Processing**, v. 140, p. 106495, 2020. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327019307162.

SAXENA, A.; GOEBEL, K.; LARROSA, C.; CHANG, F. Cfrp composites dataset, nasa ames prognostics data repository. **NASA Ames Research Center, Moffett Field, CA**, 2015.

SBARUFATTI, C.; MANSON, G.; WORDEN, K. A numerically-enhanced machine learning approach to damage diagnosis using a lamb wave sensing network. **Journal of Sound and Vibration**, v. 333, n. 19, p. 4499--4525, 2014. ISSN 0022-460X. Available from: https://www.sciencedirect.com/science/article/pii/S0022460X14003459.

SCALEA, F. L. di; SALAMONE, S. Temperature effects in ultrasonic lamb wave structural health monitoring systems. **The Journal of the Acoustical Society of America**, Acoustical Society of America, v. 124, p. 161--174, 7 2008. ISSN 0001-4966. Doi: 10.1121/1.2932071. Available from: https://doi.org/10.1121/1.2932071.

SHANG, L.; ZHANG, Z.; TANG, F.; CAO, Q.; PAN, H.; LIN, Z. Cnn-lstm hybrid model to promote signal processing of ultrasonic guided lamb waves for damage detection in metallic pipelines. **Sensors**, v. 23, n. 16, 2023. ISSN 1424-8220. Available from: https://www.mdpi.com/1424-8220/23/16/7059.

SHAO, W.; SUN, H.; WANG, Y.; QING, X. A multi-level damage classification technique of aircraft plate structures using lamb wave-based deep transfer learning network. **Smart Materials and Structures**, v. 31, 2022. Available from: https://doi.org/10.1088/1361-665X/ac726f.

SHARMA, S.; SEN, S. One-dimensional convolutional neural network-based damage detection in structural joints. **Journal of Civil Structural Health Monitoring**, v. 10, n. 5, p. 1057--1072, 2020. Available from: https://doi.org/10.1007/s13349-020-00434-z.

SHI, L.; CHENG, B.; LI, D.; XIANG, S.; LIU, T.; ZHAO, Q. A cnn-based lamb wave processing model for field monitoring of fatigue cracks in orthotropic steel bridge decks. **Structures**, v. 57, p. 105146, 2023. ISSN 2352-0124. Available from: https://www.sciencedirect.com/science/article/pii/S2352012423012341.

SILVA, S. da. Data-driven model identification of guided wave propagation in composite structures. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 40, n. 11, p. 543, 2018. Available from: https://doi.org/10.1007/s40430-018-1462-4.

SILVA, S. da; AO, J. P.; REBILLAT, M.; MECHBAL, N. Extrapolation of ar models using cubic splines for damage progression evaluation in composite structures. **Journal of Intelligent Material Systems and Structures**, SAGE Publications Ltd STM, v. 32, p. 284--295, 10 2020. ISSN 1045-389X. Doi: 10.1177/1045389X20963171. Available from: https://doi.org/10.1177/1045389X20963171.

SIMOEN, E.; ROECK, G. D.; LOMBAERT, G. Dealing with uncertainty in model updating for damage assessment: A review. **Mechanical Systems and Signal Processing**, v. 56-57, p. 123--149, 2015. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327014004130.

SIROHI, J.; CHOPRA, I. Fundamental understanding of piezoelectric strain sensors. **Journal of Intelligent Material Systems and Structures**, v. 11, n. 4, p. 246--257, 2000. Available from: https://doi.org/10.1106/8BFB-GC8P-XQ47-YCQ0.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. **J. Mach. Learn. Res.**, JMLR.org, v. 15, n. 1, p. 1929,Äì1958, jan 2014. ISSN 1532-4435.

STANDOLI, G.; SALACHORIS, G. P.; MASCIOTTA, M. G.; CLEMENTI, F. Modal-based fe model updating via genetic algorithms: Exploiting artificial intelligence to build realistic numerical models of historical structures. **Construction and Building Materials**, v. 303, p. 124393, 2021. ISSN 0950-0618. Available from: https://www.sciencedirect.com/science/article/pii/S0950061821021516.

STASZEWSKI, W. Intelligent signal processing for damage detection in composite materials. **Composites Science and Technology**, v. 62, n. 7, p. 941--950, 2002. ISSN 0266-3538. Available from: https://www.sciencedirect.com/science/article/pii/S0266353802000088.

SU, C.; JIANG, M.; LIANG, J.; TIAN, A.; SUN, L.; ZHANG, L.; ZHANG, F.; SUI, Q. Damage identification in composites based on hilbert energy spectrum and lamb wave tomography algorithm. **IEEE Sensors Journal**, v. 19, n. 23, p. 11562--11572, 2019.

SU, C.; JIANG, M.; LV, S.; LU, S.; ZHANG, L.; ZHANG, F.; SUI, Q. Improved damage localization and quantification of cfrp using lamb waves and convolution neural network. **IEEE Sensors Journal**, v. 19, n. 14, p. 5784--5791, 2019.

SU, Z.; YE, L. Lamb wave-based quantitative identification of delamination in cf/ep composite structures using artificial neural algorithm. **Composite Structures**, v. 66, n. 1, p. 627--637, 2004. ISSN 0263-8223. Twelfth International Conference on Composite Structures. Available from: https://www.sciencedirect.com/science/article/pii/S0263822304001874.

SU, Z.; YE, L. Lamb wave propagation-based damage identification for quasi-isotropic cf/ep composite laminates using artificial neural algorithm: Part i - methodology and database development. **Journal of Intelligent Material Systems and Structures**, v. 16, n. 2, p. 97--111, 2005. Available from: https://doi.org/10.1177/1045389X05047599.

SU, Z.; YE, L. Lamb wave propagation-based damage identification for quasi-isotropic cf/ep composite laminates using artificial neural algorithm: Part i - methodology and database development. **Journal of Intelligent Material Systems and Structures**, v. 16, n. 2, p. 97--111, 2005. Available from: https://doi.org/10.1177/1045389X05047599.

SZANDALA, T. Review and comparison of commonly used activation functions for deep neural networks. **arXiv preprint arXiv:2010.09458**, 2020. Available from: https://doi.org/10.48550/arXiv.2010.09458.

TELOLI, R. de O.; SILVA, S. da; RITTO, T. G.; CHEVALLIER, G. Bayesian model identification of higher-order frequency response functions for structures assembled by bolted joints. **Mechanical Systems and Signal Processing**, Academic Press, v. 151, 4 2021. ISSN 10961216.

THIERRY, V.; BROWN, L.; CHRONOPOULOS, D. Multi-scale wave propagation modelling for two-dimensional periodic textile composites. **Composites Part B: Engineering**, Elsevier, v. 150, p. 144--156, 10 2018. ISSN 1359-8368.

TONG, T.; HUA, J.; LIN, J.; ZHANG, H. Disbond contours evaluation in aluminum/CFRP adhesive joint based on excitation recovery of Lamb waves. **Composite Structures**, v. 294, p. 115736, 2022. ISSN 0263-8223. Available from: https://www.sciencedirect.com/science/article/pii/S0263822322005128.

VANLI, O. A.; JUNG, S. Statistical updating of finite element model with lamb wave sensing data for damage detection problems. **Mechanical Systems and Signal Processing**, v. 42, n. 1, p. 137--151, 2014. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327013002987.

WAN, H.-P.; MAO, Z.; TODD, M. D.; REN, W.-X. Analytical uncertainty quantification for modal frequencies with structural parameter uncertainty using a gaussian process metamodel. **Engineering Structures**, v. 75, p. 577--589, 2014. ISSN 0141-0296. Available from: https://www.sciencedirect.com/science/article/pii/S0141029614003848.

WAN, H.-P.; REN, W.-X. Stochastic model updating utilizing bayesian approach and gaussian process model. **Mechanical Systems and Signal Processing**, v. 70-71, p. 245--268, 2016. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327015003726.

WANG, D.; HE, J.; GUAN, X.; YANG, J.; ZHANG, W. A model assessment method for predicting structural fatigue life using Lamb waves. **Ultrasonics**, Elsevier B.V., v. 84, p. 319--328, 3 2018. ISSN 0041624X.

WANG, L.; YUAN, F. G. Group velocity and characteristic wave curves of Lamb waves in composites: Modeling and experiments. **Composites Science and Technology**, Elsevier, v. 67, p. 1370--1384, 6 2007. ISSN 0266-3538.

WANG, L.; YUAN, F. G. Group velocity and characteristic wave curves of Lamb waves in composites: Modeling and experiments. **Composites Science and Technology**, Elsevier, v. 67, p. 1370--1384, 6 2007. ISSN 0266-3538.

WANG, S.; WU, W.; SHEN, Y.; LIU, Y.; JIANG, S. Influence of the pzt sensor array configuration on lamb wave tomography imaging with the rapid algorithm for hole and crack detection. **Sensors**, v. 20, n. 3, 2020. ISSN 1424-8220. Available from: https://www.mdpi.com/1424-8220/20/3/860.

WANG, X.; HILL, T. L.; NEILD, S. A.; SHAW, A. D.; KHODAPARAST, H. H.; FRISWELL, M. I. Model updating strategy for structures with localised nonlinearities using frequency response measurements. **Mechanical Systems and Signal Processing**, Academic Press, v. 100, p. 940--961, 2 2018. ISSN 0888-3270.

WILLBERG, C.; DUCZEK, S.; VIVAR-PEREZ, J. M.; AHMAD, Z. A. Simulation methods for guided wave-based structural health monitoring: A review. **Applied Mechanics Reviews**, American Society of Mechanical Engineers (ASME), v. 67, 1 2015. ISSN 00036900. Available from: https://asmedigitalcollection.ez27.periodicos.capes.gov.br/appliedmechanicsreviews/article/67/1/010803/443651/Simulation-Methods-for-Guided-Wave-Based.

WU, J.; XU, X.; LIU, C.; DENG, C.; SHAO, X. Lamb wave-based damage detection of composite structures using deep convolutional neural network and continuous wavelet transform. **Composite Structures**, Elsevier, v. 276, p. 114590, 11 2021. ISSN 0263-8223.

XU, H.; LIU, L.; XU, J.; XIANG, Y.; XUAN, F.-Z. Deep learning enables nonlinear lamb waves for precise location of fatigue crack. **Structural Health Monitoring**, v. 0, n. 0, p. 14759217231167076, 2023. Available from: https://doi.org/10.1177/14759217231167076.

XU, L.; YUAN, S.; CHEN, J.; REN, Y. Guided wave-convolutional neural network based fatigue crack diagnosis of aircraft structures. **Sensors**, v. 19, n. 16, 2019. ISSN 1424-8220. Available from: https://www.mdpi.com/1424-8220/19/16/3567.

XU, Y.; KOHTZ, S.; BOAKYE, J.; GARDONI, P.; WANG, P. Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. **Reliability Engineering and System Safety**, Elsevier Ltd, v. 230, 2 2023. ISSN 09518320.

YAMASHITA, R.; NISHIO, M.; DO, R. K. G.; TOGASHI, K. Convolutional neural networks: an overview and application in radiology. **Insights into Imaging**, v. 9, n. 4, p. 611--629, 2018. Available from: https://doi.org/10.1007/s13244-018-0639-9.

YAN, A.-M.; KERSCHEN, G.; De Boe, P.; GOLINVAL, J.-C. Structural damage diagnosis under varying environmental conditions,Äîpart i: A linear analysis. **Mechanical Systems and Signal Processing**, v. 19, n. 4, p. 847--864, 2005. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327004001785.

YAN, W. J.; CHRONOPOULOS, D.; CANTERO-CHINCHILLA, S.; YUEN, K. V.; PAPADIMITRIOU, C. A fast Bayesian inference scheme for identification of local structural properties of layered composites based on wave and finite element-assisted metamodeling strategy and ultrasound measurements. **Mechanical Systems and Signal Processing**, Academic Press, v. 143, 9 2020. ISSN 10961216.

YANG, C.; YE, L.; SU, Z.; BANNISTER, M. Some aspects of numerical simulation for lamb wave propagation in composite laminates. **Composite Structures**, v. 75, n. 1, p. 267--275, 2006. ISSN 0263-8223. Thirteenth International Conference on Composite Structures. Available from: https://www.sciencedirect.com/science/article/pii/S0263822306001449.

YANG, Z.; YANG, H.; TIAN, T.; DENG, D.; HU, M.; MA, J.; GAO, D.; ZHANG, J.; MA, S.; YANG, L.; XU, H.; WU, Z. A review on guided-ultrasonic-wave-based structural health monitoring: From fundamental theory to machine learning techniques. **Ultrasonics**, v. 133, p. 107014, 2023. ISSN 0041-624X. Available from: https://www.sciencedirect.com/science/article/pii/S0041624X23000902.

YETKIN, S.; ABUHANIEH, S.; YIGIT, S. Investigation on the abilities of different artificial intelligence methods to predict the aerodynamic coefficients. **Expert Systems with Applications**, v. 237, 2024. Cited by: 0. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85170423674&doi=10.1016%2fj.eswa.2023.121324&partnerID=40&md5=8f6b15079abe9757f1e9a484f9fa3f04.

YUAN, F.-G. (Ed.). **Structural Health Monitoring (SHM) in Aerospace Structures**. 1. ed. [S.l.]: Woodhead Publishing, 2016. 514 p. ISBN 9780081001486.

ZENG, X.; LIU, X.; YAN, J.; YU, Y.; ZHAO, B.; QING, X. Lamb wave-based damage localization and quantification algorithms for CFRP composite structures. **Composite Structures**, Elsevier, v. 295, p. 115849, 9 2022. ISSN 0263-8223.

ZHANG, H.; HUA, J.; LIN, J.; TONG, T. Damage localization with lamb waves using dense convolutional sparse coding network. **Structural Health Monitoring**, SAGE Publications Ltd, v. 22, p. 1180--1192, 3 2022. ISSN 17413168.

ZHANG, P.; FENG, Y.; BUI, T. Q.; HU, X.; YAO, W. Modelling distinct failure mechanisms in composite materials by a combined phase field method. **Composite Structures**, v. 232, p. 111551, 2020. ISSN 0263-8223. Available from: https://www.sciencedirect.com/science/article/pii/S0263822319323852.

ZHANG, S.; LI, C. M.; YE, W. Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network. **Mechanical Systems and Signal Processing**, v. 147, p. 107107, 2021. ISSN 0888-3270. Available from: https://www.sciencedirect.com/science/article/pii/S0888327020304933.

ZHANG, Z.; SUN, C. Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. **Structural Health Monitoring**, SAGE Publications, v. 20, p. 1675--1688, 6 2020. ISSN 1475-9217. Doi: 10.1177/1475921720927488. Available from: https://doi.org/10.1177/1475921720927488.

ZHAO, C.; WEN, Y.; ZHU, J.; LI, T. Localization of surface dent deformation and inter-laminated damage in cfrp laminates under low-velocity impact behavior based on multi-channel one-dimensional convolutional gated recurrent unit. **Measurement**, v. 221, p. 113503, 2023. ISSN 0263-2241. Available from: https://www.sciencedirect.com/science/article/pii/S0263224123010679.

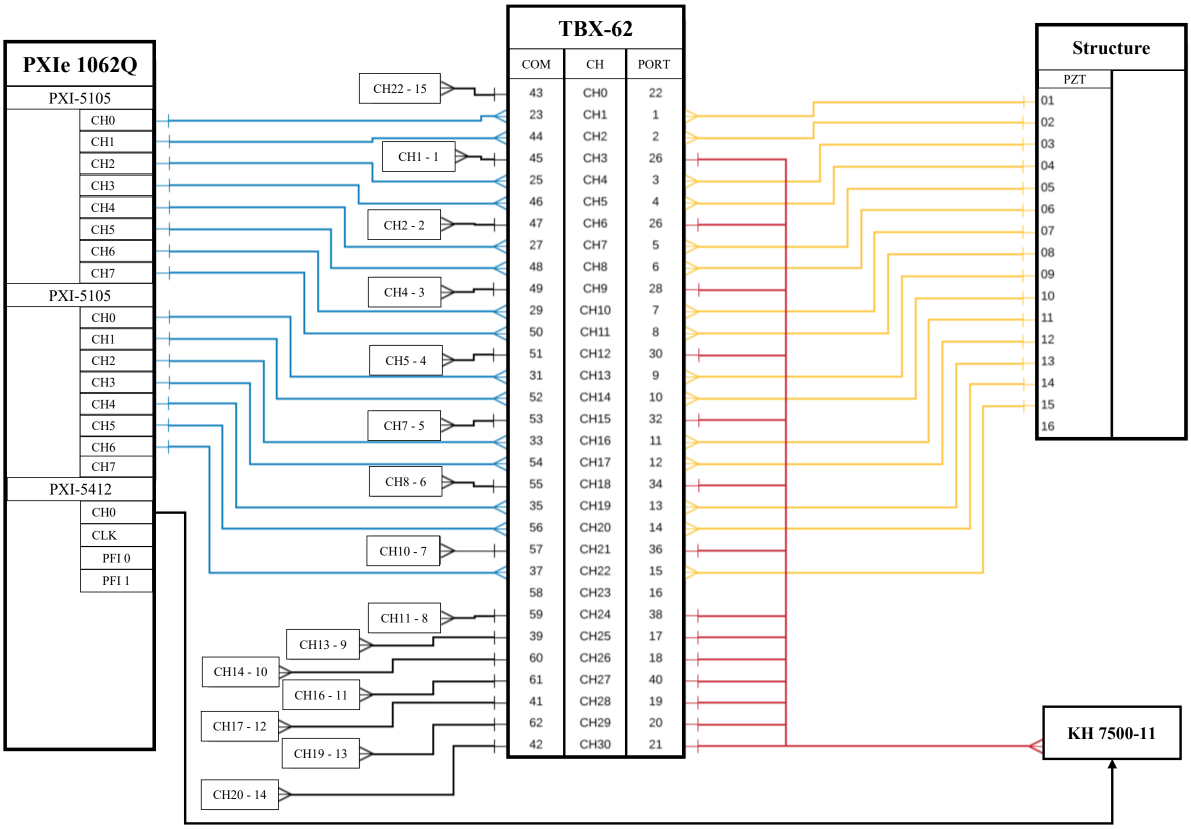
ZHAO, X.; GAO, H.; ZHANG, G.; AYHAN, B.; YAN, F.; KWAN, C.; ROSE, J. L. Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. defect detection, localization and growth monitoring. **Smart Materials and Structures**, v. 16, n. 4, p. 1208, jun 2007. Available from: https://dx.doi.org/10.1088/0964-1726/16/4/032.

ZHOU, L.; WANG, L.; CHEN, L.; OU, J. Structural finite element model updating by using response surfaces and radial basis functions. **Advances in Structural Engineering**, v. 19, p. 1446--1462, 2016. Available from: https://doi.org/10.1177/1369433216643876.

APPENDIX A – TBX-62 WIRING

The automation of the LaWaDe system was accomplished with the installation of the TBX-62 relay board connected to the NI PXI-2568 board. The TBX-62 is an interface with 31 relays that can be controlled via LabView. Instead of switching between excitation and measurement manually by interchanging probe tips among the cables, with the relay board's introduction, all switching can be automated. Figure A.1 displays the physical system's connection diagram. The relay board facilitates the connection between the PXIe measurement channels, the PZTs, and the power amplifier.

The switching system directs the input signal to the excitation PZT and connects the measurement PZTs to the PXIe. For instance, for a measurement where the first piezoelectric sensor acts as the exciter, the amplifier is connected to it, and the other sensors are linked to the PXIe. Observing the connection diagram in Figure 36, Relay Channel 3 (CH3) is activated, while channels 6, 9, 12, 15, 18, 21, 24, 25, 26, 27, 28, 29, and 30 are deactivated. Consequently, only the first PZT is linked to the amplifier. Subsequently, Relay Channel 1 (CH1) is deactivated to shield the PXIe from the high voltage excitation signal, and channels 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, and 20 are activated to link PZTs 2 to 14 to the measurement system. The PXIe's configuration has 16 measurement channels, yet only 14 are connected to piezoelectric sensors. Channel 15 has a 1:10 attenuation added and consistently receives the excitation signal in all tests, while channel 16 gets the temperature sensor signal. This configuration has enabled system automation, with switching managed through the NI Switch application integrated into LabView. Automation has dramatically expedited the process, slashing the average test time with five averages from 40 mitutes to approximately 5 minutes and significantly reducing the possibility of operator errors.



COM	CH	PORT
43	CH0	22
23	CH1	1
44	CH2	2
45	CH3	26
25	CH4	3
46	CH5	4
47	CH6	26
27	CH7	5
48	CH8	6
49	CH9	28
29	CH10	7
50	CH11	8
51	CH12	30
31	CH13	9
52	CH14	10
53	CH15	32
33	CH16	11
54	CH17	12
55	CH18	34
35	CH19	13
56	CH20	14
57	CH21	36
37	CH22	15
58	CH23	16
59	CH24	38
39	CH25	17
60	CH26	18
61	CH27	40
41	CH28	19
62	CH29	20
42	CH30	21

Figure A.1 – Relay connection implemented on TBX-62. Yellow lines indicate signal wires from plate to TBX-62 and blue lines indicate singnal wires from TBX-62 to PXI-e. Red lines indicate amplified input signal. Small rectangles indicates direct connection of the COM port to the indicated CH and PORT.