

Micro-scale Damage Detection in Metals

Using Vibration Analysis and Neural Networks

Dheeraj Chettiar

Delft University of Technology

Micro-scale Damage Detection in Metals

Using Vibration Analysis and Neural Networks

by

Dheeraj Chettiar

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday, October 16, 2026.

Student number:	6134718	
Faculty:	Faculty of Aerospace Engineering, Delft	
Project duration:	December 2025 – October 2026	
Thesis committee:	Dr. Vahid Yaghoubi Nasrabadi, Dr. Prem Sundaramoorthy, Dr. Nan Yue	TU Delft, Supervisor TU Delft, Chair TU Delft, Examiner

Front Cover:	Dheeraj Chettiar. (2026). Conceptual rendering of a Laser Doppler Vibrometer based Structural Health Monitoring and Neural Network damage detection setup [AI generated image created with ChatGPT (GPT 5.5 with DALL E)].
--------------	--

Back Cover:	Dheeraj Chettiar. (2026). Mode shape obtained from modal analysis of a plate using Autodesk® Fusion 360
-------------	---

Style:	TU Delft Report Style, with modifications by Daan Zwanveld
--------	--

An electronic version of this thesis is available at <https://repository.tudelft.nl/>.



ॐ

नेहाभिक्रमनाशोऽस्ति प्रत्यवायो न विद्यते ।
स्वल्पमप्यस्य धर्मस्य त्रायते महतो भयात् ॥

*nehābhikrama-nāśo 'sti pratyavāyo na vidyate sv-alpam apy asya dharmasya
trāyate mahato bhayāt*

***Every genuine effort builds toward eventual success, even when not
immediately apparent. Persist through apparent futility, knowing that nothing
is wasted.***

- Bhagavad Gita 2.40

Preface

In the aerospace industry, where cost is the name of the game, every moment an aircraft or a component sits idle, waiting to be inspected for damage, is a loss. This project was an attempt to see if reducing this loss and speeding up health monitoring is possible by fusing relatively simple forms of data collection with the advanced pattern recognition capabilities enabled by Neural Networks.

What initially drew me to this topic was its accessibility. Many SHM workflows rely on specialized equipment and/or access to parts of a structure that aren't easily reachable. If, by reading only the surface response of a component, we could determine whether it was damaged and subsequently pinpoint areas with a high likelihood of damage, that would change the equation entirely, making inspection viable even in settings where access to expensive, specialized equipment is challenging.

I would be lying if I said this journey was all sunshine and roses. Having no prior experience with neural networks, I struggled for a long time trying to understand how they worked, and it took a lot of trial, error, and frustration before I saw anything resembling a decent result. But along the way, I learned what mattered more than any single result: you keep working, even when progress feels invisible. Rome wasn't built in a day, and neither was this project.

I owe a great deal of thanks to my supervisor, Dr Vahid Yaghoubi, who helped me make sense of neural networks when I barely knew where to start. He was always willing to sit down and talk through a problem, and on the days I stopped believing in myself, he kept pushing me forward.

I'd also like to thank Yunhyeok Han, who helped me with much of the testing and stayed cheerful through it all; Sasan Salmani Pour Avval, whose input helped me streamline my approach to designing the neural network; and Afonso Costa, who made the tedious testing sessions genuinely fun.

To my friends, thank you for sticking with me for these past two years filled with many ups and downs and for putting up with me talking about vibrations and Neural Networks more than anyone should have to.

To my mother, thank you for believing in me and telling me to take it one step at a time when I had all but given up. And to my late father, I hope this makes you proud Daddy.

Dheeraj Chettiar
Delft, October 2026

Abstract

In a world where time is money, a structural health monitoring (SHM) technique with the ability to non destructively detect damage can be a real game changer, this work aims to take the first step in that direction. This thesis looks at two approaches: one population based and the other non-population based. Both using vibration data collected from the surface of a sample to train a neural network and plot a damage visualization, respectively.

A numerical dataset was created using finite element code for a rectangular metallic coupon, with damage being represented as localized reductions in stiffness at randomly distributed locations. The response was captured only from the top surface of each sample to emulate data collection using a scanning laser doppler vibrometer (SLDV). The neural network was trained on only a small subset of the sensor locations on the surface, while the non-population approach used all the surface sensors. For the neural network, feature selection was done using Least Angle Regression(LAR) to reduce the amount of data being processed.

The network was trained on 300 samples, with which the best performing detection model achieved about 96% accuracy despite having a sparse sensor setup. In the non-population approach, a uniform load surface[73] (ULS) based approach was used, which shows peaks and valleys near damage sites[38]. To allow any user to use the workflow easily, a graphical user interface (GUI) was created to put all aspects of data processing, neural network training and ULS visualization in one place.

The workflow proposed in the thesis looks to use NNs with data from only a few measurement points to detect damage in structures without needing to disassemble complex parts or even take them out of service. This approach could be very valuable for additively manufactured components, where conventional inspection methods like X-ray computed tomography (CT) are both time consuming and costly in terms of equipment and labour. The ULS based approach is suggested as a complementary step, which can be performed once the structure is deemed damaged to further ascertain the damage location using a denser measurement.

This work is limited in its scope to only the numerical part of this workflow, such that all results presented have been obtained only using simulated data. Therefore, some future work should investigate how well the synthetically learned knowledge transfers to the real world.

Keywords: Non-destructive, Damage Detection, Neural Networks, Frequency Response Function, Uniform Load Surface, Structural Health Monitoring, Laser Doppler Vibrometer, Mode Shapes, Convolutional Neural Networks, Kolmogorov-Arnold Networks, Sparse Sensing

Contents

Preface	i
Abstract	ii
List of Figures	vii
List of Tables	viii
I Project Introduction	2
1 Introduction	3
2 Literature Review	6
2.1 Structural Health Monitoring (SHM)	6
2.1.1 Damage in structures and models	7
2.2 Non Destructive Vibrational Analysis	8
2.2.1 Frequency Based Methods	9
2.2.2 Mode Shape Based Methods	9
2.2.3 FRF Based Methods	11
2.3 Machine Learning & Non Destructive Analysis	12
2.3.1 Feature Selection & Dimensionality reduction	12
2.3.2 Interpretability & Explainability	13
2.3.3 Deep Learning for SHM	13
2.3.4 Convolutional Neural Networks (CNNs)	14
2.3.5 Advanced Neural Network Structures	14
2.3.6 Kolmogorov-Arnold Networks	15
3 Research Questions	16
II Research Activities	20
4 Methodology	21
4.1 Finite Element Model	21
4.2 Damage Modelling	23
4.3 Frequency Response Function Generation	24
4.4 Uniform Load Surface Analysis	25
4.5 Training the Neural Network	27
4.5.1 Input to the Neural Network	27
4.5.2 Training the CNN + MLP Network	28
4.5.3 Training the CNN + KAN Network	29
4.5.4 Training the KAN Network	30
4.5.5 Training the CNN Network - Waterfall Graph	31
4.5.6 Neural Networks for damage detection on smaller plate	33
4.5.7 Analysis metrics	33

5	Results & Analysis	35
5.1	Neural Network Results	35
5.1.1	Results - CNN + MLP	35
5.1.2	Results - CNN + KAN	37
5.1.3	Results - KAN	39
5.1.4	Results - CNN (Waterfall Graph)	40
5.1.5	Neural Network Result from smaller plate	44
5.2	ULS Results	46
5.2.1	Effect of Damage Size	46
5.2.2	Effect of Damage Severity	48
5.2.3	Effect of Mesh Size	49
III	Conclusions and Future Work	52
6	Conclusion	53
6.1	Research Question outcomes	53
6.1.1	Research Question 1	54
6.1.2	Research Question 2	55
7	Future Work	56
7.1	Limitations	56
7.2	Future Work Recommendations	56
7.2.1	Experimental Validation	57
7.2.2	Advanced Damage Modelling	57
7.2.3	Larger Structures	57
7.2.4	Sensor Configuration	57
7.2.5	Damage Localization	57
7.2.6	Neural Network Architectures	58
7.2.7	Sensing Techniques	58
7.2.8	Generalization	58
IV	Introduction to the GUI	60
8	The GUI	61
8.1	Why a GUI?	61
8.2	The Main Window	61
8.3	ULS Viewer	61
8.4	Node Picker	62
8.4.1	FRF Viewer	62
8.4.2	Waterfall Graph Viewer	62
8.5	Neural Network Trainer	62
V	Appendix	70
A	GUI Images	71

List of Figures

1.1	Complete SHM workflow	3
1.2	X Ray CT visualization of defects in a part	4
2.1	SHM Flow	7
2.2	Types of damage seen in metals [68]	8
2.3	Types of vibration sensors: Accelerometer, Laser Doppler Vibrometer, PZT sensor (Left to right)	9
2.4	Natural frequency shift	9
2.5	(a) Mode shape of a beam, (b) Mode shape curvature of beam, (c) MAC matrix	10
2.6	Representation of ULS curvature [73]	11
2.7	Constituent parts of an FRF. Top Left: Magnitude, Top Right: Phase, Bottom Left: Real part, Bottom Right: Imaginary part	11
2.8	Comparing classical approach vs deep learning based approach [6]	14
2.9	Overview of machine learning paradigms and commonly used neural network architectures.	15
2.10	Typical difference between neural network(a) and KAN(b) [15]	15
4.1	Comparison of FRFs from a single sensor of the different mesh sizes a - Mesh 3, b - Mesh 2 and c - Mesh 1	22
4.2	Magnitude and Phase plot for comparing FRFs of healthy and damaged samples	24
4.3	Real and Imaginary plot for comparing FRFs of healthy and damaged samples	25
4.4	Comparing healthy and damaged ULS curvature to visualize damage	26
4.5	Performance of the different FRF components across some of the datasets	27
4.6	Locations of sensors used for neural network training	27
4.7	Basic network architecture of CNN + MLP network with residual and attention mechanisms BN - Batch Normalization, GAP - Global average pooling	29
4.8	Basic network architecture of CNN + KAN network GAP - Global average pooling	30
4.9	Basic network architecture of KAN network	30
4.10	Representation of magnitude plot of FRF with equivalent waterfall plot	31
4.11	Representation of stitched waterfall image used as input for simple 1D CNN	31
4.12	Basic network architecture of 1D CNN BN - Batch Normalization, GAP - Global average pooling	32
4.13	Representation of stacked waterfall image used as input for simple 2D CNN	32
4.14	Basic network architecture of 2D CNN BN - Batch Normalization, GAP - Global average pooling	33
4.15	Location of sensors for the small plate used for neural network training	33
5.1	Balanced accuracy and AUC trends for the CNN + MLP network Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	36
a	Balanced Accuracy	36
b	AUC	36
5.2	Sensitivity and specificity trends for the CNN + MLP network Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	36

a	Sensitivity	36
b	Specificity	36
5.3	Accuracy plot for the CNN + MLP network to show overfitting trends	37
a	Accuracy - Config_1	37
b	Accuracy - Config_2	37
5.4	Accuracy plot for the CNN + MLP network to show overfitting trends	37
a	Accuracy - Config_3	37
b	Accuracy - Config_4	37
5.5	Balanced accuracy and AUC trends for the CNN + KAN network Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	38
a	Balanced Accuracy	38
b	AUC	38
5.6	Sensitivity and specificity trends for the CNN + KAN network Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	38
a	Sensitivity	38
b	Specificity	38
5.7	Accuracy plot for the CNN + KAN network to show overfitting trends	39
a	Accuracy - Config_1	39
b	Accuracy - Config_2	39
5.8	Accuracy plot for the CNN + KAN network to show overfitting trends	39
a	Accuracy - Config_3	39
b	Accuracy - Config_4	39
5.9	Balanced accuracy and AUC trends for the KAN networks Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	40
a	Balanced Accuracy	40
b	AUC	40
5.10	Sensitivity and specificity trends for the KAN network Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	40
a	Sensitivity	40
b	Specificity	40
5.11	Balanced accuracy and AUC trends for the 1D CNN network with waterfall graph Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	41
a	Balanced Accuracy	41
b	AUC	41
5.12	Sensitivity and specificity trends for the 1D CNN network with waterfall graph Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	41
a	Sensitivity	41
b	Specificity	41
5.13	Balanced accuracy and AUC trends for the 2D CNN network with waterfall graph Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	42
a	Balanced Accuracy	42
b	AUC	42
5.14	Sensitivity and specificity trends for the 2D CNN network with waterfall graph Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue	42
a	Sensitivity	42

b	Specificity	42
5.15	SHAP importance plot for 2D CNN network on Config_4	43
5.16	SHAP importance plot for 1D CNN network on Config_4	43
5.17	SHAP importance plot for CNN + MLP network on Config_4 selected through LARS	44
5.18	Performance figures of different networks on the small plate dataset with damage size 0.5mm and severity 60 (Config_1)	44
5.19	Performance figures of different networks on the small plate dataset with damage size 0.75mm and severity 80 (Config_4)	45
5.20	ULS plots of Mesh 1 (a,b) and Mesh 3 (c,d) with three damage points, showing the impact of damage size.	47
a	Mesh 1 - Size 0.5 mm, Severity 80	47
b	Mesh 1 - Size 0.75 mm, Severity 80	47
c	Mesh 3 - Size 0.5 mm, Severity 80	47
d	Mesh 3 - Size 0.75 mm, Severity 80	47
5.21	ULS plots of Mesh 2 with one damage point, showing the impact of damage severity.	48
a	Mesh 2 - Size 0.75 mm, Severity 60	48
b	Mesh 2 - Size 0.75 mm, Severity 80	48
5.22	ULS plots of Mesh 3 with one damage point, showing the impact of damage severity.	49
a	Mesh 3 - Size 0.75 mm, Severity 60	49
b	Mesh 3 - Size 0.75 mm, Severity 80	49
5.23	ULS plots of Mesh 1, Mesh 2 and Mesh 3 with three damage points, showing the impact of mesh size.	50
a	Mesh 1 - Size 0.75 mm, Severity 80	50
b	Mesh 2 - Size 0.75 mm, Severity 80	50
c	Mesh 3 - Size 0.75 mm, Severity 80	50
A.1	Main window of the GUI	71
A.2	ULS Viewer app	72
A.3	Main screen of the Node Picker app	73
A.4	FRF viewing screen of the Node Picker app	74
A.5	Waterfall graph viewing screen	75
A.6	CNN Trainer app	76
A.7	Result Plotter app showing Multi-Metric plot	77
A.8	Result Plotter app showing ROC curve plot	77
A.9	Result Plotter app showing Learning curves plot	78
A.10	Result Plotter app showing Confusion matrix plot	78

List of Tables

3.1	Summary of Research Questions	17
4.1	Damage configuration for neural network dataset generation	23
5.1	Damage configuration for non-population approach	46
6.1	Research Question 1	54
6.2	Research Question 2	55

Part I

Project Introduction

1

Introduction

Structural Health Monitoring (SHM) as a field has been around for a long time[69, 18, 3, 13], constantly upgrading and updating itself as time passes. As newer materials are developed and innovative manufacturing techniques[52] are introduced, the challenge of assessing their structural integrity in a non-destructive manner arises. Any damage left undetected for long periods runs the risk of degrading performance or, in the worst cases, total failure. SHM's basic goal is to detect damage in a given specimen, and over time, the ability to locate the damage, assess damage severity and suggest best remedies have also been brought under the SHM umbrella[60].

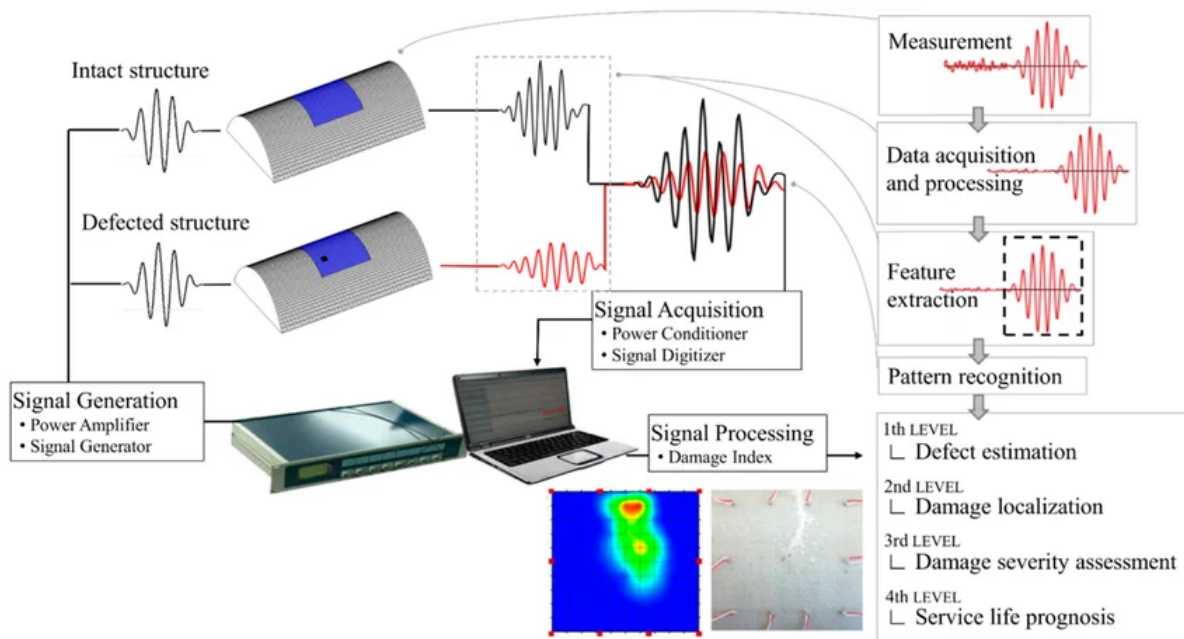


Figure 1.1: Complete SHM workflow

While internal damage to parts can be easily seen upon destructive testing, the unique challenge for SHM is to be able to do it in a non-destructive manner. For the purpose of non-destructive damage detection (NDDD), a plethora of both contact and non-contact techniques have been developed and successfully tested for various applications, ranging from large struc-

tures like buildings and bridges[1, 25] to smaller structures like individual components or parts used in machines.

The techniques can range from seemingly simple to exceptionally complex. The field began by using simple vibrational responses from the sample[25] and slowly moved on to combining the response with the input vibrations to generate the Frequency Response Function (FRF)[16], from which metrics such as mode shapes, mode shape curvatures, etc., could be extracted through simple mathematical manipulations. Over time, more complex solutions like ultrasonic testing were introduced[56], which were more effective at pinpointing the exact location of the damage but had the downside of being material dependent[34]. Perhaps, at the very top sits X-ray Computed Tomography (CT), which has the ability, in essence, to look into the material using high powered X-rays just like the ones used for medical imaging. The downside of this approach is that the costs of the machine and the expertise to run it are very high, along with a lot of time being needed for high resolution imaging.

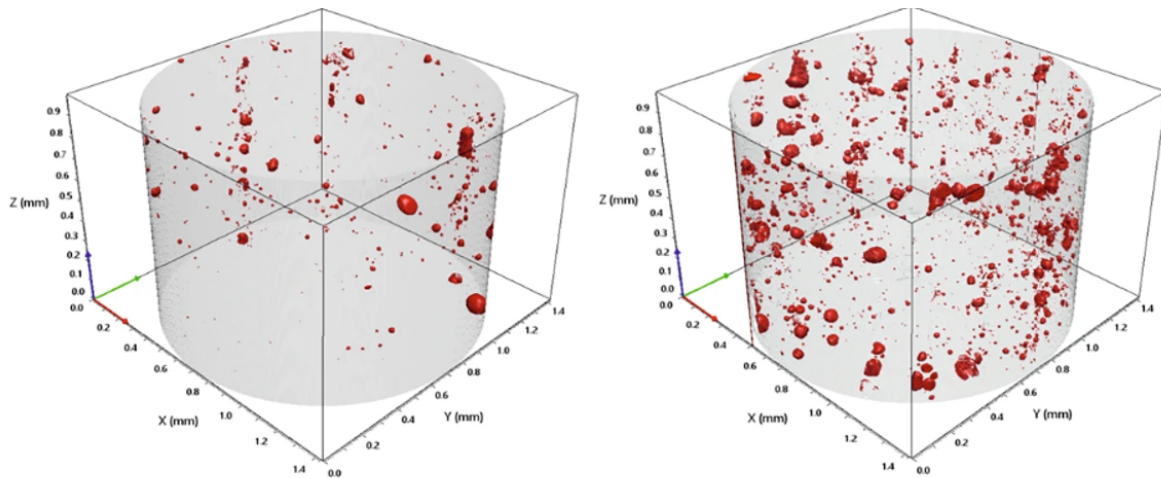


Figure 1.2: X Ray CT visualization of defects in a part

As the world has learned more and more about Artificial Intelligence (AI), Neural Networks (NN) and Machine Learning (ML), there has been increased interest in using their abilities in pattern recognition to aid in damage detection and subsequent processes[36, 72]. ML has found many applications in real-time monitoring of structures to predict damage before it occurs, as well as in general damage detection, due to its ability to work with diverse data types, such as FRFs, mode shapes, images, etc.

One popular aspect of SHM/NDDD that researchers have often targeted is how few sensors can be used to get the job done[55]. This is a valid question, as more sensors often mean more measurement time, and in some cases, it might not be practical to attach many sensors to the part because they might interfere with the body's dynamics or there just isn't enough space. This aspect can also benefit from ML, as it might be able to pick up on trends even with a smaller amount of data.

This thesis proposes a two-pronged approach. One population based approach working on training a neural network using FRFs from a sparse sensor network to determine whether a given sample is damaged or not. The 2nd, a non-population based approach working on a Uniform Load Surface (ULS)[73] based indicator using a dense sensor network. FRF data is only collected from the top surface of a FEM model so as to emulate the use of a Scanning

Laser Doppler Vibrometer (SLDV), all work is based on simulations on coupon sized samples and aims to be a stepping stone for experimental validation on real components.

2

Literature Review

This chapter gives an overview of the literature reviewed during the course of this project, so as to provide a basic understanding of relevant concepts. The literature covers structural health monitoring, damage in metallic structures, vibration based damage detection, machine learning approaches for SHM and the use of neural networks in damage detection.

2.1. Structural Health Monitoring (SHM)

Structural Health Monitoring (SHM) is often defined as the process of implementing a damage detection strategy for aerospace, civil and mechanical applications [26]. Detecting damage in structures and SHM in general has been around since the 70s [3, 13, 17], and the field has seen a resurgence since the 2000s [70, 74, 25], where early works concentrated on offshore oil platforms, the current boom focuses on different sensing techniques, vibration analysis, as well as advanced data analysis methods.

To segment the vast field of SHM into clear, bounded and achievable tasks, Rytter suggested a four level system [60]. The process was divided into;

- **Level 1: Detection** - Determining whether the specimen is damaged.
- **Level 2: Localization** - Once damage is detected, determining its location in the specimen.
- **Level 3: Assessment/Quantification** - Determining the severity of impact of the damage on the specimen.
- **Level 4: Prediction/Prognosis** - Determining the remaining useful life of the specimen

The basis of SHM is the fact that damage alters the dynamic properties of the structure, which we can ascertain from measuring its response [13]. When these damages are in early stages, i.e. they are small in size, their effect on the structure's response also remains small. The problem arises when the changes caused by damage fall within the same range as those caused by environmental changes and measurement noise [64, 47].

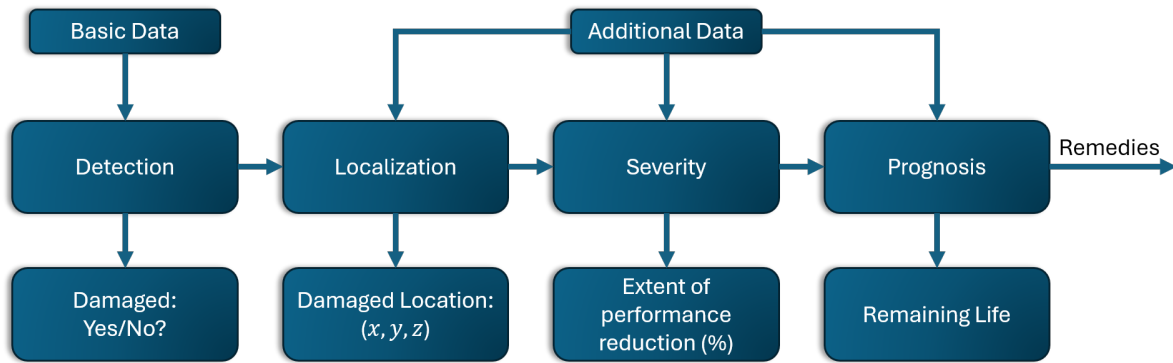


Figure 2.1: SHM Flow

To detect said damages, many different techniques have been used over time, from simple acts like listening to the sound made by the machine, to using specialized sensors like accelerometers, Laser Doppler Vibrometers(LDV), Impedance Sensors, Fiber optic Sensing, etc. [30]. While the sensors enabled the collection of many different data points, damage could only be detected by analyzing them, for which, over time, many different techniques were developed to transform the data into damage indicators[61, 53, 41]; recently, with the advent of neural networks(NNs), a push towards using raw data rather than engineered indicators has also been seen [72].

Apart from all these sensors and techniques, another factor is the type of damage being studied. Different types of damage can arise from different mechanisms and affect the structure slightly differently, like fatigue cracks from repeated loading, corrosion from environmental factors or general wear and tear damage.

2.1.1. Damage in structures and models

Structures can degrade through many different types of damage, such as fatigue cracking, corrosion, plastic deformation, delamination and other local material degradation [39, 8, 33]. Metallic structures are more prone to certain types of damage, like fatigue cracks due to cyclic loading, surface/subsurface corrosion due to environmental conditions, plastic deformation due to impacts or operating beyond design loads and even defects that might occur during the manufacturing process [24]. These damages, while originating from different mechanisms, alter the local mechanical properties, which in turn affect the local stiffness and load carrying capacity as they grow; potentially leading to failure [59]. They can also end up changing the dynamic response of the structure in a subtle/not so subtle manner.

From a SHM modelling point of view, the reason/mechanism for the damage is not as important as the effect the damage has on the structure's properties and dynamic behaviour. Damage can be modelled very accurately with realistic modelling approaches, but if the goal is to study the change in global behaviour, a simplified model of the damage can suffice [11]. One such simplification is to represent the damage using a local reduction of stiffness [37, 14, 54]. While this method can't capture things like crack tip stress concentrations, crack opening and closing, etc., it does provide the necessary information about local stiffness while reducing computational requirements.

Most work on detecting damage in structures, has concentrated on larger structures with damage considered small relative to the structure's size. Very little work is done on small/micro-scale damage detection, especially using non contact methods.

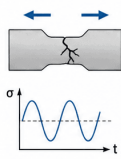
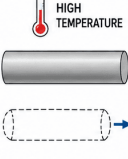
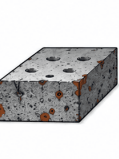
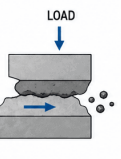
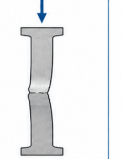
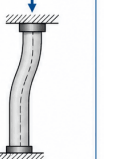
	FATIGUE	CREEP	CORROSION	WEAR	OVERLOAD	BUCKLING
						
 PRIMARY DRIVER	Cyclic loading	High temperature	Environment	Sliding contact	Excessive load	Compression
 TYPICAL CLUE	Crack growth	Slow deformation (strain over time)	Pitting or rust	Surface loss (wear marks)	Sudden deformation	Lateral instability
 PREVENTION	Reduce stress concentration	Lower stress or better alloy	Coating or material selection	Lubrication or harder surface	Increase capacity	Bracing or geometry

Figure 2.2: Types of damage seen in metals [68]

2.2. Non Destructive Vibrational Analysis

As stated earlier, many different techniques were developed over time to assist in damage detection. Early techniques like acoustic signatures, while intuitive, were not always reliable and also required a human to be in the loop, which led to the development of many nondestructive testing (NDT) methods [35]. As computation capacity increased, modern techniques based on finite element modelling (FEM) were created, which enabled much more realistic simulations of the structure. In all these cases, the vibrational response signals of the structure were used to derive damage indicators which were much more interpretable than the raw signal.

The basic premise behind vibration based damage detection is that the dynamic response of the structure changes based on its geometry, mass, stiffness, damping, etc. [13]. When damage alters these properties, studying the change to the dynamic response in a non destructive manner via damage indicators can allow us to identify damaged specimens from healthy ones without needing to break them [18].

While there are many vibration analysis methods of particular interest due to widespread use are:

- **Frequency Based Methods:** Oldest and simplest methods, measuring quantities like frequency shifts and ratios.
- **Mode Shape Based Methods:** Giving more spatial context to the structure instead of just numbers.
- **Frequency Response Function Based Methods:** Ratio of input and output carries a lot of information.

An advantage of vibration based methods is the fact that measurements do not necessarily have to be taken directly at the damage location; damage in one region can affect the response of another region [67]. Measurements can be obtained using both contact sensors, like accelerometers, or non-contact sensors, like Laser Doppler Vibrometers(LDV), each with its own pros and cons. The number and placement of measurement locations can also vary from dense to sparse, depending on the available space and measurement time, all of which can affect the accuracy.



Figure 2.3: Types of vibration sensors: Accelerometer, Laser Doppler Vibrometer, PZT sensor (Left to right)

2.2.1. Frequency Based Methods

Natural frequency of a structure is among the simplest dynamic characteristics used for damage identification [61, 40]. Since the natural frequency of a structure depends on its stiffness, mass and other mechanical properties, any change in these can result in shifts in one or more natural frequencies. Damage indicators can therefore be created from the frequency shift of individual modes, combinations of modes, or ratios between natural frequencies [13].

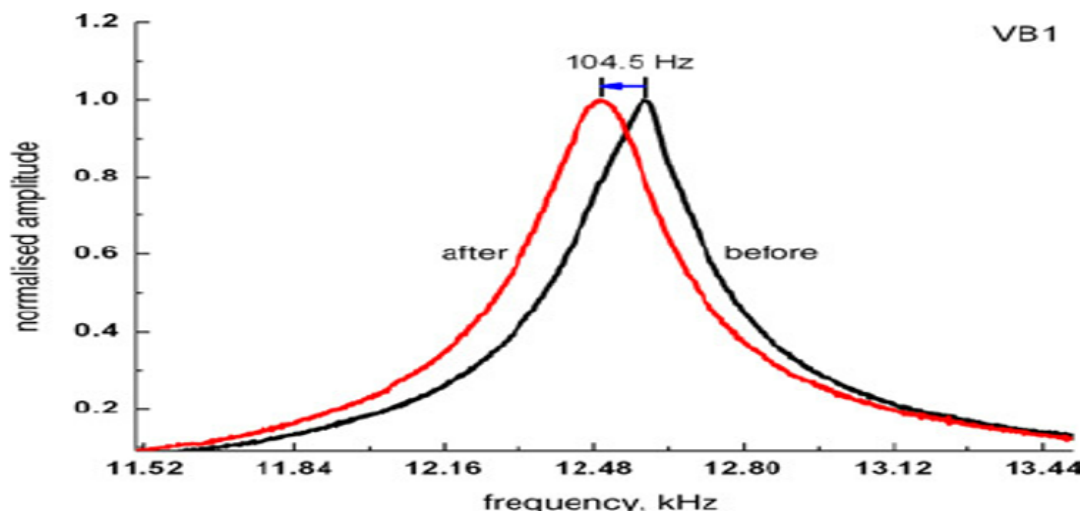


Figure 2.4: Natural frequency shift

In cases where the damage is very small, the resulting shift could also be small enough to not be distinguished from measurement noise or from variations in measurement conditions [61].

2.2.2. Mode Shape Based Methods

While the natural frequency tells us about the frequency at which the structure vibrates, it does not give any information regarding how the structure deforms. Mode shape is the frequency dependent deformation pattern which describes how each point on the structure moves relative to the others [23].

The spatial nature of mode shapes makes them very useful for damage detection, as comparing the mode shapes of a healthy and damaged structure can provide information about the presence of damage. Based on mode shapes, several damage indicators have been devel-

oped, such as mode shape curvature, Modal Assurance Criterion (MAC), Co-ordinate Modal Assurance Criterion (COMAC), and many more [28].

Mode shape curvature based on the spatial derivative of the mode shape has been shown to be a good damage indicator, as small changes in mode shape can create large changes in its curvature [41]. The MAC identifies damage by comparing the similarity between mode shapes, where less than ideal similarity could indicate damage [53]. COMAC expands this by considering the contribution of individual measurement locations to the whole mode shape [41].

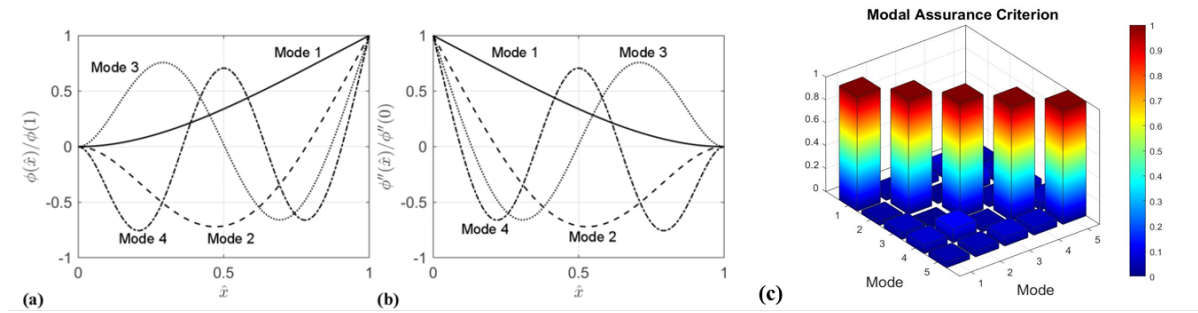


Figure 2.5: (a) Mode shape of a beam, (b) Mode shape curvature of beam, (c) MAC matrix

The downside of these mode shape based methods is that accurate spatial information requires a large number of measurement locations. Methods related to spatial derivatives/curvatures can be very sensitive to measurement noise; all these issues might amplify when trying to identify small damages with a sparse network of sensors.

2.2.2.1. Uniform Load Surface

While individual mode shapes can provide some information about the deformation of the structure, combining many of them together can provide much more information regarding the damage. One such approach is Uniform Load Surface (ULS), which was developed for damage detection and localization, where the response of a structure to a uniformly distributed load is studied[73].

The effectiveness of the ULS can further be enhanced by using its spatial derivative to highlight local changes, which is analogous to the mode shape and mode shape curvature. One more way the ULS and derived indicators could be improved is by using a healthy baseline to compare against, which can help pick out the minute differences [73], especially when a larger number of samples isn't available to make a population based approach possible. In this study, the ULS based method is considered as a complementary method to the population based approach.

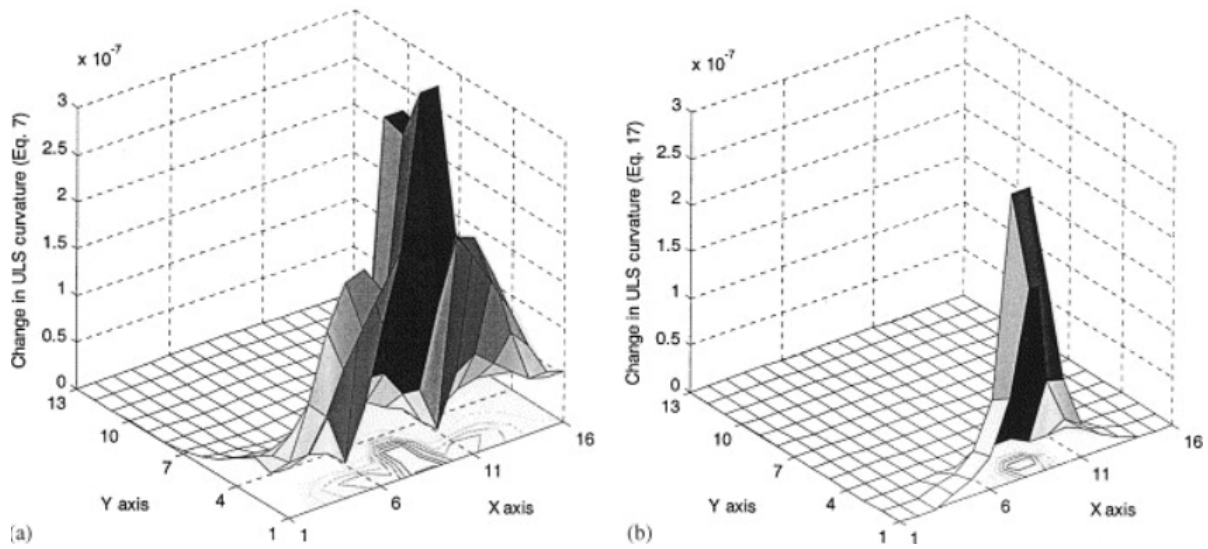


Figure 2.6: Representation of ULS curvature [73]

2.2.3. FRF Based Methods

An alternative to using individual parameters is to work directly with frequency response functions (FRFs), which are frequency dependent relations between the applied input and the observed structural response [71]. FRFs contain information over a range of frequencies, and can also be used to find other modal characteristics like natural frequencies and mode shapes [23]

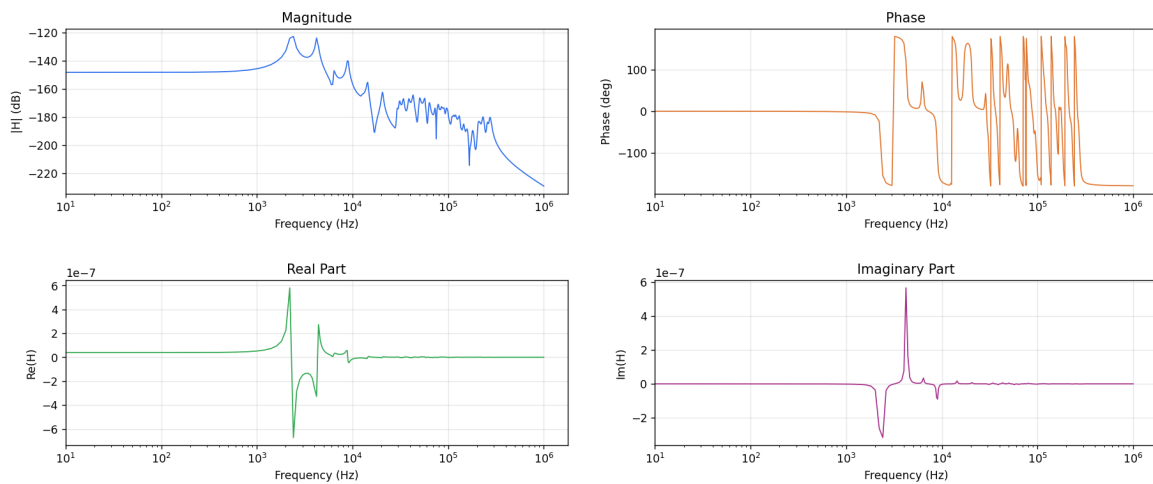


Figure 2.7: Constituent parts of an FRF. Top Left: Magnitude, Top Right: Phase, Bottom Left: Real part, Bottom Right: Imaginary part

FRFs retain a lot more of the data than any extracted modal parameter. Bundling this dense information with measurements from multiple locations can provide lots of information about the response of different parts of the structure across different frequencies. Changes caused by small damage could therefore show up in the response of multiple locations as small deviations, allowing for easier detection.

While the large amount of data in FRFs makes them a good choice, they also suffer from

high dimensionality. An FRF measured over a very fine frequency resolution could contain hundreds of thousands of data points; coupled with data from multiple locations, the data size blows up pretty quickly and not all of these data points will necessarily carry useful information. Feature selection and dimensionality reduction techniques can be used to address this challenge to make the data more suitable for machine learning purposes.

FRFs allow us to step away from using predefined damage indicators, which may or may not capture the desired information. Rather than cutting down the data, the complete frequency response could be used for data driven damage detection. Machine Learning models are particularly well suited for this task as they can learn the complex relationships that exist between the response and the damage state by exploiting the dense data of the FRF [27].

2.3. Machine Learning & Non Destructive Analysis

For a very long time, the non destructive damage identification problem was solved by creating a mathematical relation between the data and the damage state, but by reframing the problem as a pattern recognition problem [72, 27], a more data driven approach can be taken in which various machine learning models and algorithms can be used. The ML models can much more easily learn the relationships between structural responses and damage states, which is paramount as the use cases become more and more complex and hard to distinguish.

Early application of ML in SHM used damage sensitive features extracted from the complete response and used them as inputs to different classifiers [32] or regression models [66]; these features included natural frequencies, frequency shifts, mode shapes, etc., as stated earlier. The early ML models, like Support Vector Machines (SVM) [22], Random Forest [10], k-nearest neighbours and linear regression, have shown very good performance at mapping response to damage state [4].

One aspect that demands attention is the fact that vibration measurement data contains a large amount of information with many potentially useful variables, but of these, only a few may contain information relevant to finding the damage. In such a case, feeding extra information to the model might negatively impact its performance; as such, it would make sense to select on a few relevant features or reduce the dimensionality of the data.

2.3.1. Feature Selection & Dimensionality reduction

Feature selection aims to pick out from the given data a subset of features best suited for the task at hand. While many approaches have been created over the years, Least Angle Regression (LARS) is rather well suited for the damage detection task. LARS chooses features most strongly correlated with the residual and gradually adjusts its coefficients till another feature becomes equally correlated [20]. This "subsampling" can help reduce the complexity of the subsequent learning problem.

Dimensionality reduction, on the other hand, takes a different approach by transforming the current feature space onto a new, smaller set of variables. Principal Component Analysis (PCA) is one such popular method. PCA finds the orthogonal directions (principal components) which capture the largest amount of variance. The original data can then be projected onto these components, thus leaving us with a lower dimensional representation of the original data [49, 63].

The difference between feature selection and dimensionality reduction is of great importance. Feature selection retains a subset of the original data, thus also retaining their physical mean-

ing, while dimensionality reduction makes new variables which don't hold onto the old information. For the same data, their outputs could be very different, and this aspect can change how well an ML model can learn.

2.3.2. Interpretability & Explainability

With the increasing complexity of machine learning models and deep learning networks, the term "black box" is more often mentioned to signify the fact that, as humans, we cannot comprehend the predictions made by the machine. Models and networks which can show in human understandable terms which parts of the data they relied on to make their decision are termed as interpretable [19, 44].

While interpretability aims to design inherently understandable networks, explainability is asking a black box model what more information it can provide [57, 58]. One such explainable method is SHAP, which gives information about the importance of certain parts of data to the final outcome [46]. In the case of non destructive vibration analysis, SHAP can help analyse if the damage is more easily visible in lower or higher frequency bins. With increasingly deeper and complex networks being designed to tackle complicated tasks, interpretability and explainability can play an important role in data processing as well as the quality of the model outcome.

2.3.3. Deep Learning for SHM

The classical approaches to machine learning, as described above, rely heavily on feature extraction for prediction; Features first need to be extracted from measured data, after which the model can use them to distinguish between healthy and damaged structures. Deep learning reduces this dependence on extracted features by using neural networks to learn many such hidden features directly from the data [21, 12].

An Artificial Neural Network (ANN) is an ML model consisting of computational units(neurons) arranged in layers. each neuron takes its input and modifies it using certain learnable parameters and linear/non-linear transformations; This allows the network to learn complex relationships between input data and desired output [48, 42, 6]. During training, the difference between the predicted and desired result is used to tune the network through various optimization algorithms [29].

A key benefit of using NNs for non destructive damage detection is their ability to learn the nonlinear relations in the data. This is particularly useful as damage may create small changes across many frequencies and measurement locations rather than a large change in a single location. Being able to piece all the information together directly from the data rather than relying on manually defined indicators is why deep learning is seeing more and more use in SHM and non destructive damage detection.

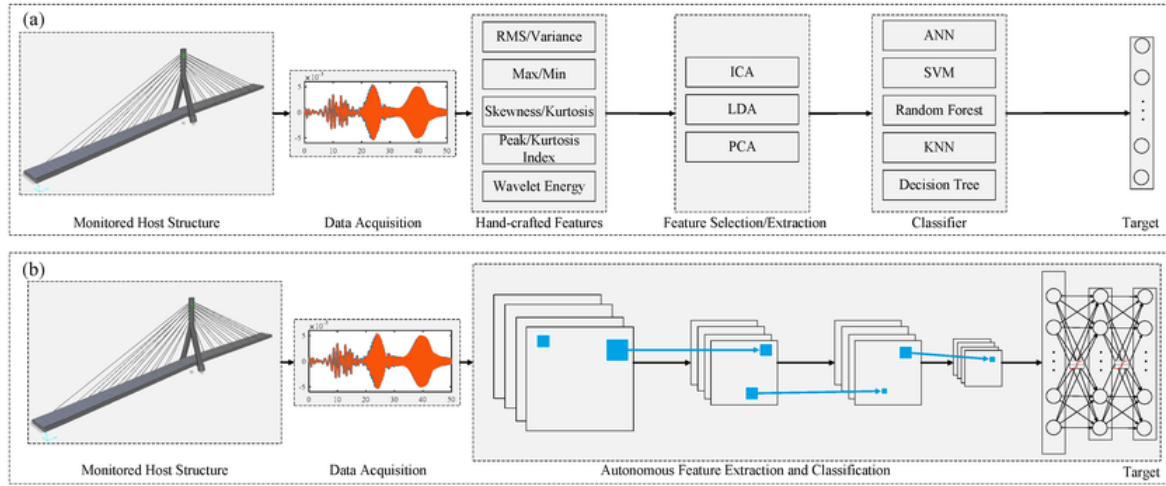


Figure 2.8: Comparing classical approach vs deep learning based approach [6]

2.3.4. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are an extension of ANNs, which can use convolutional filters to learn recurring patterns from a local region of the input [5, 9]. CNNs were originally made for image recognition, where neighbouring pixels have relational information, but have since been used for various tasks like face recognition, pose estimation, etc [9]. Beyond image based work, they have also found uses in damage detection by using various structural response data where neighbouring frequencies can also carry relational information [50, 2]. Particularly useful for SHM is the fact that response data can be arranged according to frequency and measurement location to provide more structure to the data, which can be better understood by the CNN.

Nowadays, neural networks are being used for more and more complex tasks, and there comes a point when simply adding more layers to the network does not necessarily improve performance and might even make the model difficult to train. As such, different specialized architecture have been developed to help alleviate these issues.

2.3.5. Advanced Neural Network Structures

Instead of looking at neural networks as a sequence of identical layers, many new approaches augment layers to improve how the network understands the data under not so ideal scenarios. Two approaches are of particular interest, namely residual learning and attention mechanisms.

2.3.5.1. Residual Networks

Residual networks are modification applied to CNNs which allows the input data to take a shortcut through a layer [31]. What this means is that apart from processing the data through the layer, the input is also passed along to the output so that the network can learn information using the residual between the processed and raw data [62]. This helps combat degrading performance and vanishing gradients in deep and complicated networks.

2.3.5.2. Attention Mechanism

While the input data can contain many regions of interest, not all of them carry the same amount of information about the damage. Attention mechanism gives the network the ability to weight different parts of the input according to how relevant it is to the task at hand. This is

particularly useful when damage information is concentrated in a few clusters within the data [43, 51, 75]. This ability to essentially sort through the learned features might prove useful in picking up the subtle changes that small/micro-scale sized damage can cause.

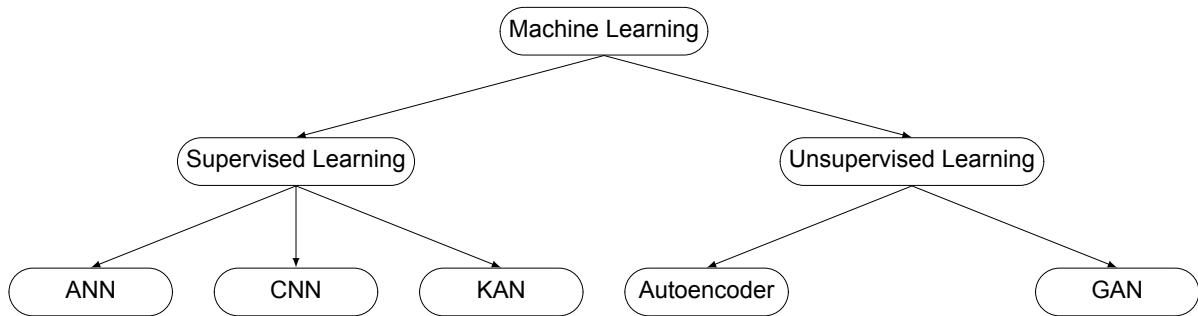


Figure 2.9: Overview of machine learning paradigms and commonly used neural network architectures.

2.3.6. Kolmogorov-Arnold Networks

A pretty recent development in deep learning architectures is that of Kolmogorov-Arnold Networks (KAN), which work on the Kolmogorov-Arnold representation theorem, which essentially states that any multivariate continuous function can be represented as a finite composition of univariate functions [7].

In a standard multi-layer perceptron (MLP), the connections between the neurons are represented as learnable weights, while nonlinear functions are used as activation functions in the neuron; in KANs, the connection between neurons is represented as learnable functions, commonly parametrized as splines. This allows for better learning of nonlinear relations. [45, 65]

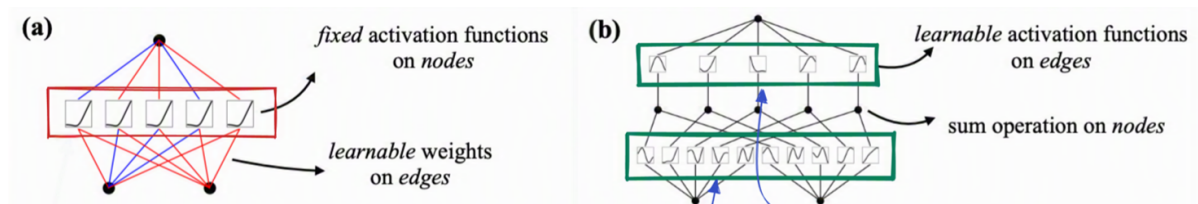


Figure 2.10: Typical difference between neural network(a) and KAN(b) [15]

3

Research Questions

This chapter looks at the research gaps that exist pertaining to the detection of small/micro-scale damage in metals, particularly focusing on the use of neural networks and vibration analysis and formulates research questions.

While a lot of research has been done on damage detection over the years using both data driven and vibration analysis approaches, certain specific approaches could still benefit from further exploration

1. **Limited investigation of small/micro-scale damage:**

- Although vibration based damage detection has been widely studied for structures like bridges, beams and plate, many studies only consider damage that is relatively large and easily observable at or near the surface.
- The detection of small/micro-scale subsurface damages, where the changes in structural response are considerably weaker remain less explored

2. **Sensitivity of conventional vibration based damage indicators:**

- A wide range of damage indicators based on natural frequency and mode shapes have been developed, however, their sensitivity to small/micro-scale damage is not well known.

3. **Neural network performance under sparse sensing conditions:**

- Deep neural networks are increasingly being used for SHM purposes, often using large datasets with dense sensor layouts.
- Less attention is paid to shallow network especially when training samples are limited and sensor network is sparse.

Based on these points and the literature reviewed, a main research question was formulated with various sub question with the aim of addressing some of the identified gapes.

Main Research Question: How can vibration data be used for detecting small/micro-scale damage in metals?

This question sparks many other more specific questions, which are discussed below.

The first point that comes up is the fact of how many samples are needed to detect the damage; can a solution be developed where just by studying a single sample, a distinction can be made or, does there need to be many samples, both damaged and healthy to help figure out the state. This tells us that both a population and a non-population based approach should be looked at.

Next, focusing on the population based approach, the natural question that arises is what neural network is best suited for the task at hand, and how their performances compare to each other. Apart from the type of network, does the type of data being used also affect the performance and what exactly in the sample determines how well any network performs.

For the non-population approach, is it possible to detect the damage from just one sample or is a healthy baseline important. What factors affect the ease of detecting the damage, and how important is each of the identified factors relative to each other.

Research Questions		
Main RQ: How can vibration data be used for detecting small/micro-scale damage in metals?		
RQ1	Can a population based approach be used to detect small/micro-scale damage using vibration data?	
	RQ 1.1	How do different network architectures perform at the detection task?
	RQ 1.2	What effect does the formatting of the input data have on the performance?
	RQ 1.3	What factors affect the network's performance?
RQ 2	Can a non population based approach be used to detect small/micro-scale defects using vibration data?	
	RQ 2.1	Which metric is best suited for detecting the damage?
	RQ 2.2	How does the size of damage affect the ease of identifying it?
	RQ 2.3	What other factors affect the ease of detecting the damage?

Table 3.1: Summary of Research Questions

Part II

Research Activities

4

Methodology

This chapter describes the methodology used to investigate the detection of micro-scale damages in coupon sized metal plates using their vibration responses. Two complimentary methods are investigated. The first, a population based approach using neural networks trained on the FRFs from both healthy and damaged samples. The second, a non-population based approach, in which damage is identified by comparing the vibration derived indicators of a damaged sample to a reference healthy sample.

The methodology has four main parts. First, a finite element model of the plate is developed to generate its vibration response. Next, damage is introduced to the plate by a localized reduction in Young's modulus, allowing samples with different damage counts, damage sizes, and damage severities to be generated. Next, the FRF are generated by using modal properties after excitation is applied, and these are used for both the population based and non-population based approaches. Finally, different neural networks are trained using a sparse sensor setup to study the effect of network architecture on damage detection.

4.1. Finite Element Model

A FE model of the two layer plate was created to generate the vibration data needed for this study. The model was created using Python's scikit-fem library, and the scipy library's sparse eigenvalue solver was used to solve the model's modal properties. A 3D formulation was used to capture the through thickness and out of plane vibration responses. This approach allows the damage to be placed at different positions in the plate's volume, including subsurface regions.

The plate is modelled as two layers with slightly different material properties. Although additive manufacturing is not investigated directly in this work, the layered configuration was chosen a forward looking modelling choice, allowing the damage to be evaluated in regions with different material properties and providing a potential base for further work.

The plate has a length of 100 mm, a width of 25 mm and a total thickness of 4 mm. Both layers are assigned the same nominal material properties values of $E = 162$ GPa, $\nu = 0.278$ and $\rho = 8440$ kg/m³. For every sample, the values of the material properties are randomly varied $\pm 3\%$, to introduce variation between samples, even the healthy samples. This variation is intended to reduce the possibility of the neural network associating a single sample's response as healthy rather than learning from its other characteristics.

The plate is discretized using eight node trilinear hexahedral elements. The datasets for neural network training had a mesh of 20 elements in the X-direction, 10 elements in the Y-direction and 24 elements in the Z-direction. This mesh size is maintained for all samples used for training the neural network, so that any changes in the response are due to material and damage and not due to mesh variations.

A separate mesh study was done for the vibration analysis part of the study, 3 different mesh sizes were used to see the effect of mesh size; the only change in those meshes is in the X and Y directions. The three meshes end up being:

- 1. 10x5x24 elements - **Mesh 1**
- 2. 20x10x24 elements - **Mesh 2**
- 3. 30x15x24 elements - **Mesh 3**

The plate is kept in FREE-FREE condition, with no other boundary constraints. The modal solution, therefore, has 6 rigid body modes, which are excluded before synthesizing the FRF. While the FRF is evaluated from 10 Hz to 1 MHz, the relatively large in plane size of the mesh elements means that higher frequency responses may not be represented very well. This can also be seen in the frequencies that the network chooses to majorly focus on as shown in the results, which are in the range up to 200 kHz.

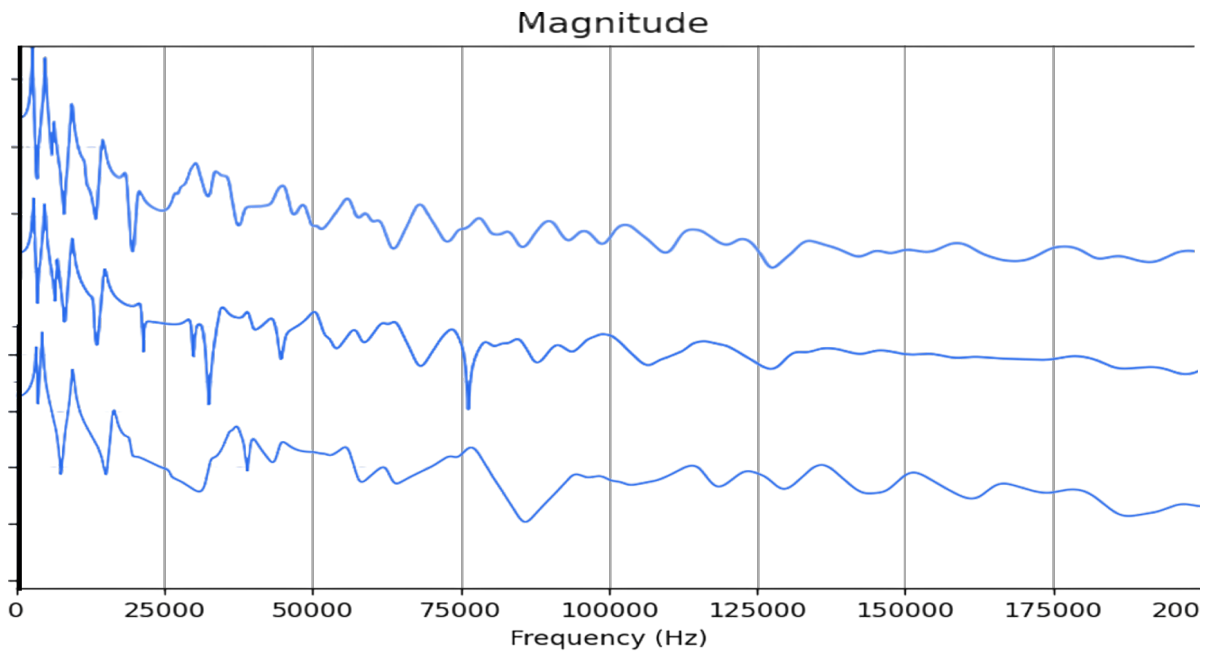


Figure 4.1: Comparison of FRFs from a single sensor of the different mesh sizes
a - Mesh 3, b - Mesh 2 and c - Mesh 1

From the above plot, we can see that mesh sizing is very important for capturing a proper response from the structure. Of the three meshes, Mesh 1 performs very poorly by not being able to capture all the data; Mesh 2 and Mesh 3 perform much closer to each other, with Mesh 3 obviously performing better, particularly in the region around 30 to 40 kHz. A trade-off decision had to be made between accuracy and computation time; as such, Mesh 2 was chosen to keep computation time manageable for the large datasets, while keeping in mind that using finer meshes may net better results.

Apart from the plate described above in the methodology, another smaller plate with its length limited to 25 mm instead of the current plate's 100 mm was also tested to see how some of the networks performed on it. A mesh of 10x10x24 was used for this plate to investigate if the smaller size meant that damage could be detected more easily, since more of the plate is technically damaged. All other parameters were kept the same, except for the length of the plate.

4.2. Damage Modelling

In this study, damage has been represented as a localized reduction in the Young's modulus of an element rather than by explicitly modelling it as a change in geometry, thus avoiding the need for remeshing of the sample every single time. This simplification of the damage makes it less computationally intensive and allows for generating datasets much quicker; it also means that any damage configuration can be created with ease.

Each damage is defined by its 3D location (x, y, z), a damage diameter and a damage severity. In this study, two damage diameters are used, 0.5 mm and 0.75 mm, so as to look at the performance of the damage indicators and the neural networks when damage size increases. The damage severity is the term that controls how much the Young's modulus reduces by; here, two values, 60% and 80% reduction, have been used to again study the difference in performance.

Once the Young's modulus is reduced, the stiffness reduction is applied using a Gaussian spread. The local reduction value is given as:

$$\delta(\mathbf{x}) = \beta \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_d|^2}{2\sigma^2}\right) \quad (4.1)$$

Where β is the damage severity (0.6 or 0.8), $\mathbf{x}$ is the centroid of the element being evaluated, and $\mathbf{x}_d$ denotes the location of the damage, while σ controls the spread. Once the local reductions are calculated, the reductions are combined multiplicatively for all the damages influencing a single point, with no change to the density or Poisson's ratio. The Gaussian distribution allows for a spatially distributed damage region instead of representing the damage as a single point of change.

Name	Description	Damage Size (mm)	Damage Severity (%)	Damage Count
Config_1	Size50_Sev60	0.5	60	0 to 5
Config_2	Size50_Sev80	0.5	80	0 to 5
Config_3	Size75_Sev60	0.75	60	0 to 5
Config_4	Size75_Sev80	0.75	80	0 to 5

Table 4.1: Damage configuration for neural network dataset generation

Each generated sample has either no damage present, implying a healthy sample or between 1 and 5 damages. The damage count is randomly chosen to create a number of different scenarios with varying damage counts and locations. The positions of the damages are randomly distributed within the volume of the plate, while maintaining some margins from the side, top

and bottom faces to make sure that the damage remains completely subsurface. Additionally, a distance of 5 mm is also enforced between the centres of each damage to maintain separation and to avoid multiple damage distributions from overlapping.

The same damage modelling and variation was also applied on the smaller 25 mm plate, as well, with datasets being limited to Config_1 and Config_4 to see the performance and the extremes.

4.3. Frequency Response Function Generation

Once the altered Young's modulus values due to damage are calculated, the mass and stiffness matrices are assembled using the sample's material properties. The generalized eigenvalue problem is then solved to obtain the modal characteristics.

The resulting values are then used to calculate the frequency response function using modal superposition, which avoids needing to solve a separate dynamic solution at every excitation frequency, making computations faster over a larger frequency range. A constant modal damping of $\zeta = 0.02$ is used to further simplify calculations, using which the FRFs are constructed and stored as complex values so as to carry both the magnitude and phase information along with the real and imaginary parts, so as to have more options to choose from when training the neural networks.

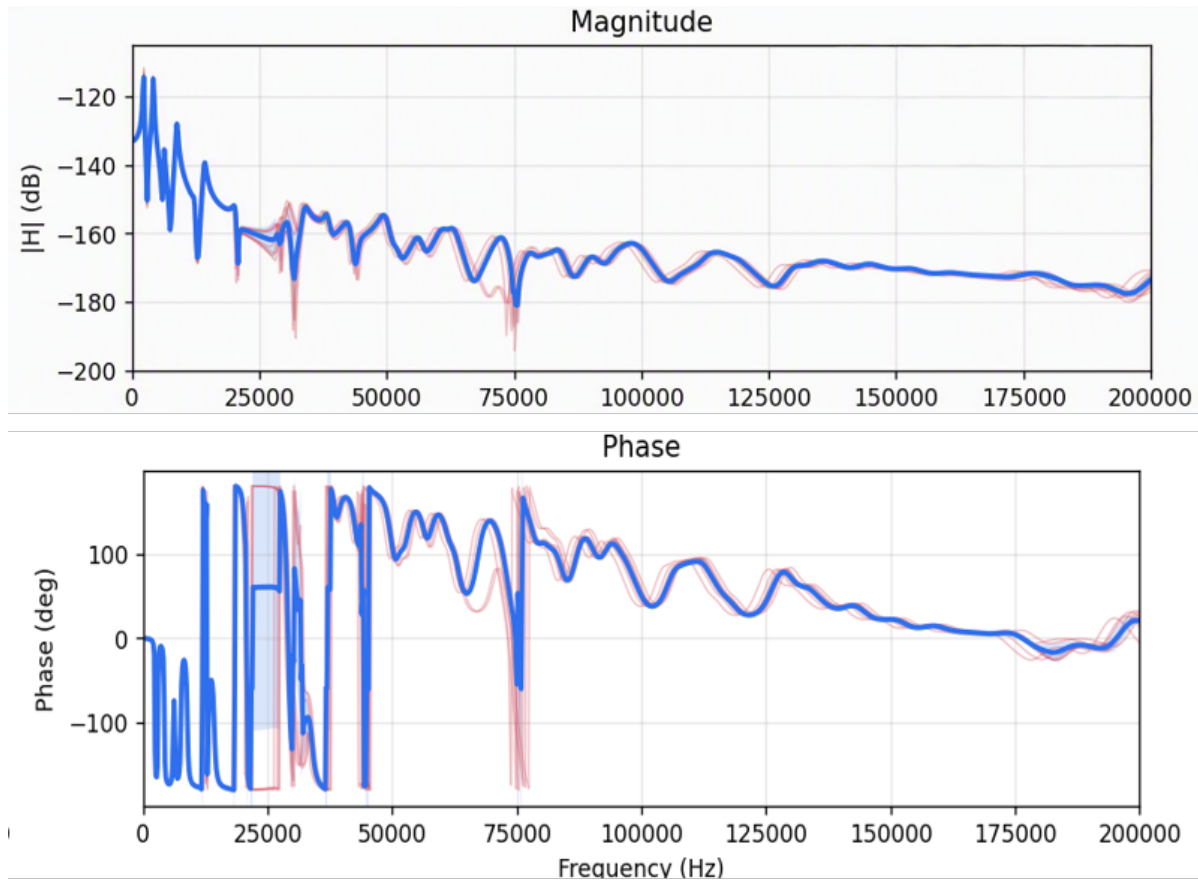


Figure 4.2: Magnitude and Phase plot for comparing FRFs of healthy and damaged samples

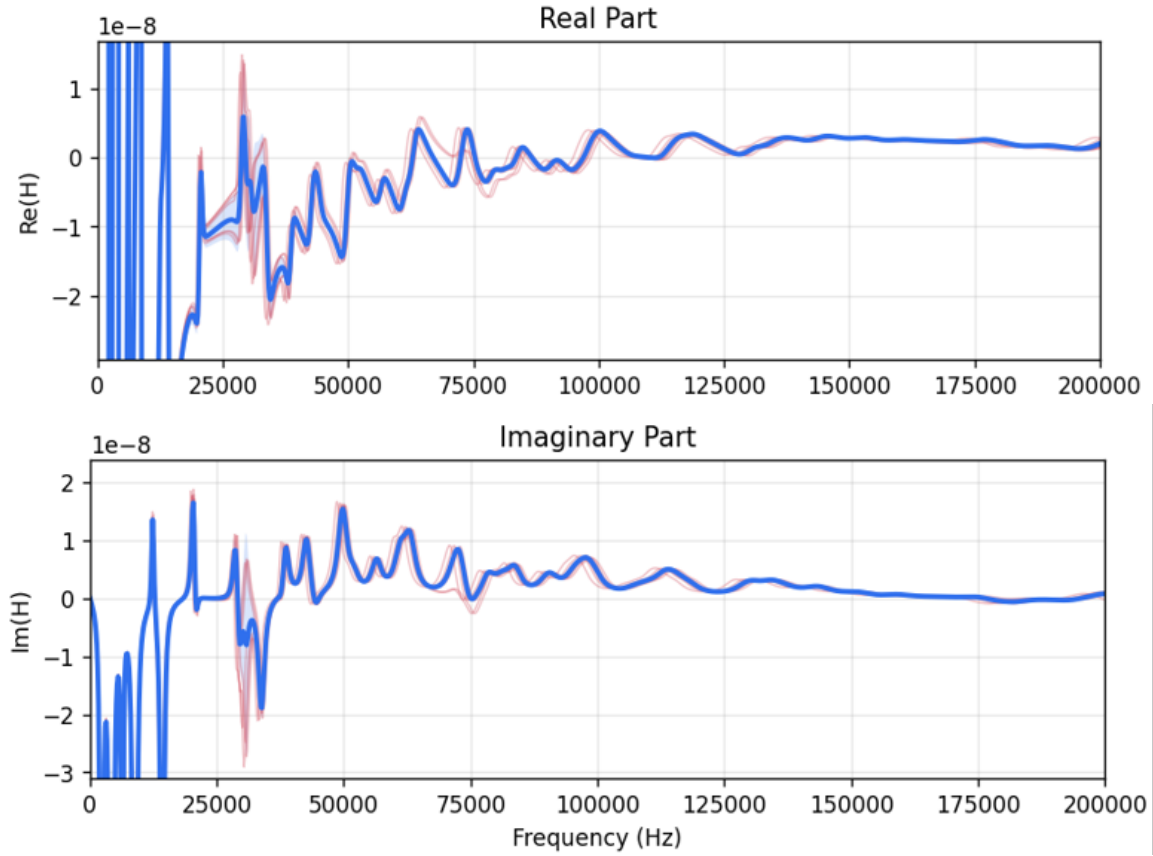


Figure 4.3: Real and Imaginary plot for comparing FRFs of healthy and damaged samples

A frequency range from 10 Hz to 1 MHz is sampled using 6000 frequency bins to capture the response at both low and high frequencies, as different parts may carry important information for the neural network to learn from. A fixed excitation is applied to all samples to make sure no other variables come into play. The excitation is applied on the bottom surface of the plate close to the bottom right corner in an out of plane direction.

The response to the input is only captured at nodes located on the plate's top surface. In the current configuration (Mesh 2), the top surface contains 231 measurement nodes from which only a select subset will be chosen as input for training the neural networks, allowing the effect of sparse sensing to be evaluated. The complete response of the top surface is still preserved for the non-population approach. The restriction of reading only the top surface is to try and emulate the use of a Scanning Laser Doppler Vibrometer (SLDV), where the vibration response is measured from one surface while excitation is applied on another.

4.4. Uniform Load Surface Analysis

The Uniform Load Surface (ULS) based approach was implemented as a complementary approach to the population based approach. Unlike the neural networks, which require a large number of healthy and damaged samples to learn from, the ULS method compares the vibration based indicators of a damaged sample to a reference healthy sample.

The ULS method works by using modal data extracted from the generated FRFs from the entire top surface of the plate. It combines contributions of multiple modes to create a displacement

field for a uniformly distributed load. This differs from mode shapes and mode shape curvature, where only one mode is considered at a time.

The modal information is picked from the generated FRFs through peak picking, and the detected frequencies are then used to create the mode shapes from the stored complex FRFs at all the surface nodes. Each mode shape is normalized to ensure that visualizations stay consistent. For a selected mode r , the ULS displacement contribution (w) is:

$$w_r = \frac{\phi_r \Gamma_r}{\omega_r^2} \quad (4.2)$$

$$\Gamma_r = F \sum_{j=1}^N \phi_{jr} \quad (4.3)$$

Here, ϕ_r is the normalized mode shape, while ω_r is the corresponding frequency and Γ_r is the modal participation factor under the applied uniform load. An additional option was also investigated, where a weighted change was made to ϕ_{jr} , replacing it with ψ_{jr} , to try and increase the participation of the higher modes.

$$\psi_{jr} = \phi_{jr}(a + b\omega_r + c\omega_r^2) \quad (4.4)$$

Where a, b and c are coefficients to control the relative contribution of the terms; if a is set to 1 and b and c are set to zero, the equation goes back to being the standard implementation. For a selected mode range, the displacements are first summed to get the ULS displacement field, and then the curvature is calculated from this, whose absolute value is used as the ULS curvature.

To identify the damage, the ULS curvature is compared to the ULS curvature of a healthy sample. In places where the two samples differ, the plots peaks and valleys indicate the possible presence of damage. In certain cases where this ULS curvature (damaged - healthy) indicator cannot pick up the damage, another indicator is used, which takes the above mentioned indicator and compares it to the mode shape curvature over the selected mode range.

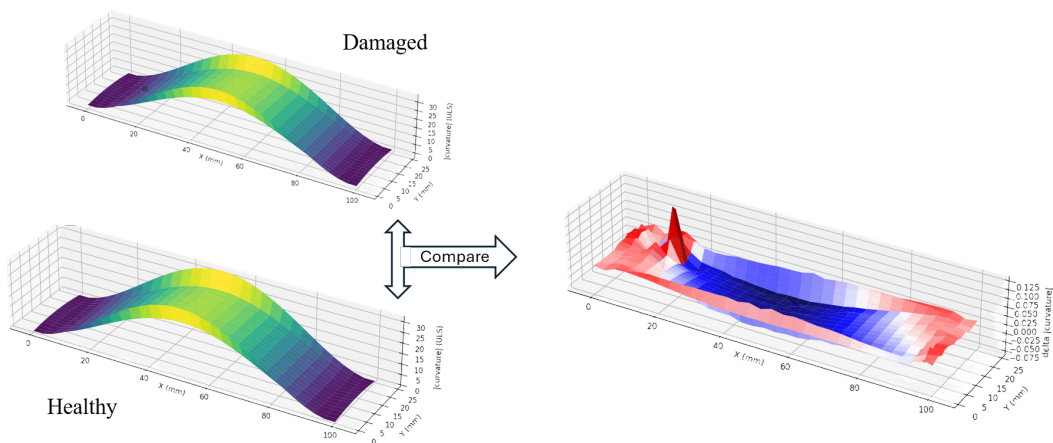


Figure 4.4: Comparing healthy and damaged ULS curvature to visualize damage

To study the effect of different mesh sizes, damage sizes and severities, a separate dataset was generated which looped through all the different mesh, damage size and damage severity configurations but kept the damage locations constant so that a proper comparison could be made between them.

4.5. Training the Neural Network

4.5.1. Input to the Neural Network

The FRFs generated in the previous steps were used as the main input for training the neural networks. For every measurement location, the FRFs being complex valued meant that, other than just using the real and imaginary parts of FRFs, the magnitude and phase information could also be derived to be used as inputs if needed. In most cases, it was observed that of the four components, the imaginary and real parts performed at par, and from these the imaginary part was chosen as the main input.

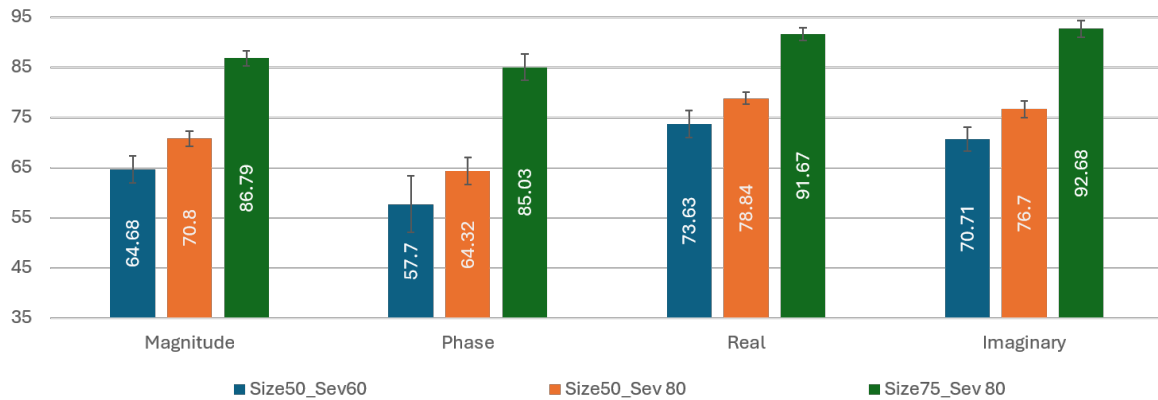


Figure 4.5: Performance of the different FRF components across some of the datasets

For all neural networks, a choice was made to use 300 samples as the training set; this number was chosen as a balance between the generally large number of samples required for training and the limited availability of samples in real industrial scenarios.

Moreover, since the goal is to study how well a sparse sensor setup performs, only four of the available 121 surface nodes were selected to be used as inputs. The four nodes were randomly selected such that each one lay in one quadrant of the plate so as to preserve some spatial information.

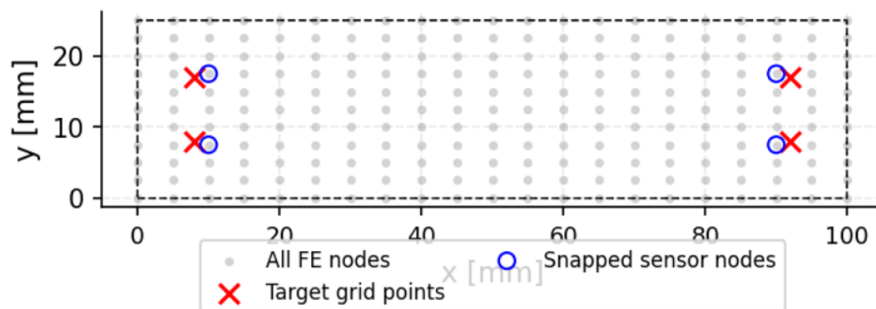


Figure 4.6: Locations of sensors used for neural network training

Despite selecting just four sensor locations, each location's FRFs contained 6000 frequency bins, out of which, upon inspection of the graphs, it could clearly be seen that roughly only the first 4000 bins contained valuable data, hence a choice was made to use only those bins for further processing. To further reduce the amount of data, feature selection was implemented through the use of Least Angle Regression (LARs). Through LARs, the top 300 bins were extracted from the available bins; only 300 bins were selected due to the large imbalance between the available frequency features and the number of samples.

From these 300 selected bins, the input to the different neural networks was constructed. The input was structured as a 3D matrix with a 2x2 face chosen so that the four sensors could be placed according to their real location in the plate's quadrant. This was done to try and preserve some of the spatial information so that the neural networks could learn from it if necessary. The 3rd dimension in this case contained the frequency bins, thus allowing the networks to stride through the frequencies to pick up relations.

The same basic input was used for training both the CNN and KAN based networks across all the different datasets, with the imaginary part of the FRF being used as the sole input. An additional approach was also attempted, where instead of using the raw FRF values, they were slightly modified into a waterfall graph by assigning colours depending on the point's place relative to peaks and valleys in the plot. Here, a simple CNN was given two different shapes of input; the first, where all 4000 bins from each sensor were stitched end to end and fed as a single 16000 long line of inputs, and the second, where the 4000 bins were stacked on top of each other to form a pseudo image; the two approaches being 1D and 2D, respectively.

4.5.2. Training the CNN + MLP Network

The first network evaluated was a Modified Convolutional neural network (CNN) with a Multi-layer perceptron (MLP) head, where the CNN deals with the deep learning and the MLP deals with the final classification using the deep learning. As described in the previous section, the input was prepared as a 3D matrix with the goal of preserving some spatial relation between the sensors.

The goal was to see how relatively shallow networks would perform at the damage detection task, so the network had only 3 convolutional blocks, where each block contained a convolutional layer, batch normalization, a rectified linear unit (ReLU) activation and max pooling. Each block has integrated into it a residual connection, which allows it to have a shortcut for the data to flow through, so that the network can better learn more from the difference between the original input and the learned information.

After each residual block, there is an attention mechanism integrated, which pools together all the sensor information and passes it through its own small fully connected network using a sigmoid activation to obtain information about which parts are more relevant to the task. This allows the network to modify and adapt the importance of different learned features. Dropouts are used in both the convolutional and fully connected layers to randomly drop certain parts of the learned representation to avoid overfitting. The three layers have 128, 64 and 16 output channels with 4, 64 and 8 as their kernel sizes, respectively.

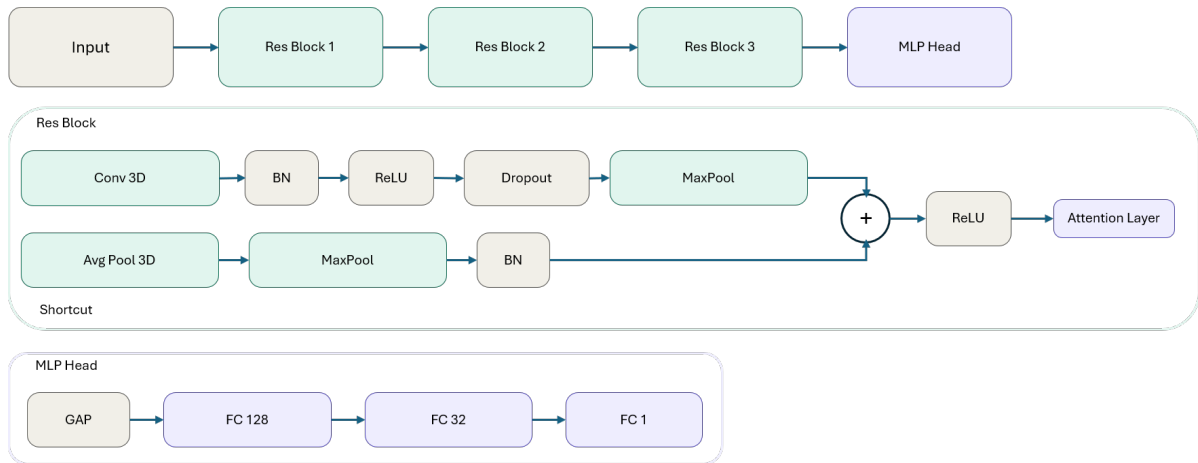


Figure 4.7: Basic network architecture of CNN + MLP network with residual and attention mechanisms
BN - Batch Normalization, GAP - Global average pooling

The extracted features were passed to the classification head with two fully connected layers outputting a single logit, as a binary classification between the healthy and damaged states. The network was trained as stated earlier, using 300 training samples for a maximum of 2000 epochs, where if the performance of the network does not improve for 500 consecutive epochs, the network enforces an early stop. An AdamW optimizer with an initial learning rate of 1e-3 and weight decay of 1e-4 was used along with a cosine annealing learning rate scheduler to allow for tuning using larger steps during the initial training and smaller steps later on. A total batch size of 8 was used to strike a balance between learning performance and computational time.

To reduce the dependence on results from a single run, the complete training process is repeated five times with random seed values. For every repeat, the test and validation sets are kept constant, with changes made only in the training set. Additionally, the training and validation sets are given a fifty-fifty split of healthy and damaged samples, while the test set has a roughly 25-75 split between healthy and damaged samples.

4.5.3. Training the CNN + KAN Network

With a traditional deep network having been tested, the next network evaluated is a slight modification on the previous CNN + MLP network, such that the MLP head is replaced with single layer KAN head and the modified CNN becomes a simple CNN, thus allowing us to evaluate how a combination of a simple deep network with a modern structure like KAN performs.

The basic change made to the previous CNN block is the removal of the residual skip learning part, as well as shifting to a shallower two layer. Just like before, the network moves through the frequency dimension of the data to learn the relations. The two layers consist of 64 and 32 output features with kernel sizes of 31 and 15, respectively. Like in the previous setup, an attention layer is placed after each convolution block to help the network learn which parts of the feature map are more important. After being processed through the convolution part of the network, the features are pooled into the singular 64 feature KAN layer, which finally gives a single value, a 0 or a 1 as its output; the KAN layers are implemented using the `efficient_kan` library, and inside the KAN layer, a spline of order 3 is used.

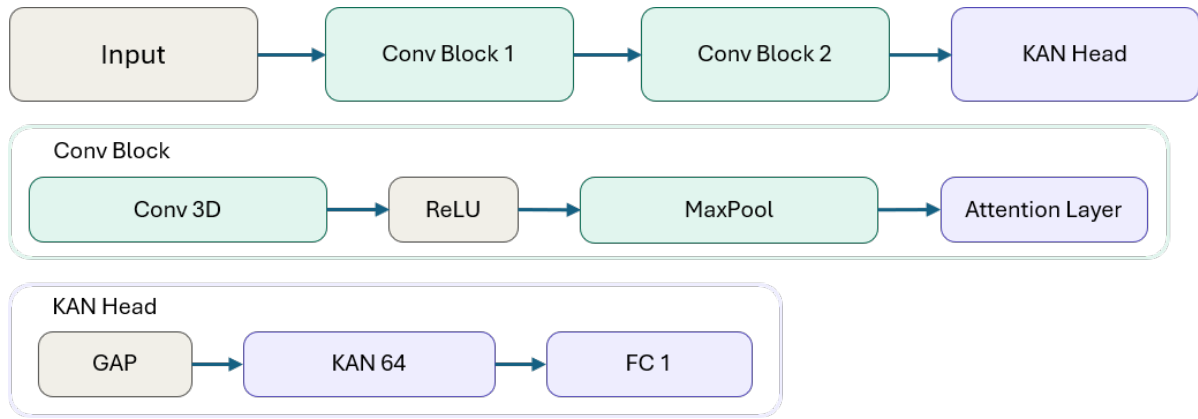


Figure 4.8: Basic network architecture of CNN + KAN network
GAP - Global average pooling

In this case as well, the input is the imaginary part of the FRF trimmed to 4000 bins, then having feature selection performed using LARS to extract 300 features to be used as input. Along with this, the AdamW optimizer is used with the base learning rate set to $1e-3$ and the base weight decay set to $1e-4$, this is coupled with a cosine annealing scheduler to help alter the learning rate and decay throughout the run, first in large steps, then in smaller ones. The complete training is done for 2000 epochs, with the early stopping activated if performance doesn't improve for 1000 consecutive epochs.

Once again, instead of a single run, performance is averaged over multiple runs, with test and validation sets being common across and the training set being remade every time. The fifty-fifty split is maintained in the training and validation sample, with a roughly 25-75 split being present in the test set.

4.5.4. Training the KAN Network

In the previous two sections, a mixture of deep learning networks and advanced networks were seen: a modified CNN with a MLP head and a simple CNN with a KAN head. The next logical step would be to test the performance of a network completely made using an advanced architecture.

The KAN network used in this section builds on the KAN head used in the CNN + KAN network, by maintaining the spline order at 3, but considerably increasing the number of layers to 8, with the following being the number of neurons in each layer: 128, 64, 32, 8, 8, 16, 32, 64; each KAN layer has just a simple dropout attached to it.

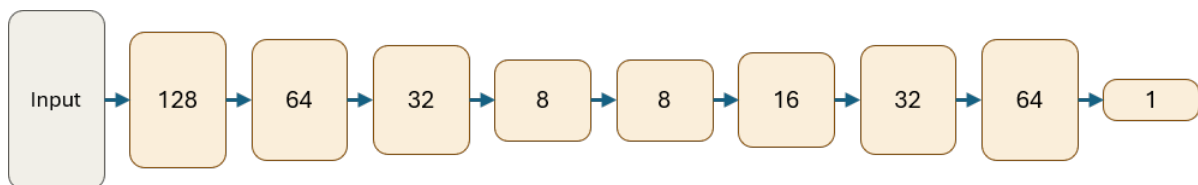


Figure 4.9: Basic network architecture of KAN network

As before, the model is trained on the imaginary part of the FRF with 4000 bins and 300 features being selected using LARs as the feature selection algorithm. The training duration is slightly increased here to a maximum of 3000 epochs, with an early stopping criterion of

2000 epochs. This is paired with the AdamW optimizer, with the learning rate and weight decay initially set to $1e-3$ and $1e-4$, respectively, with the cosine annealing scheduler used to tune those parameters.

4.5.5. Training the CNN Network - Waterfall Graph

Until this point, other than feature selection being applied to the input data, no other changes have been made to its representation. In the previous sections, many different and complicated architectures have been used; in contrast, this section sticks to a basic CNN architecture.

A rather simple modification is made to the input FRF data, where colours are assigned to every value in the plot, depending on whether it is a peak, a valley or an in-between point. This information gets stored akin to a single line of pixels. Based on this, two different approaches, the stitched waterfall graph and the stacked waterfall graph, were tried, which differed in terms of how the individual sensors' waterfall graphs were handled.

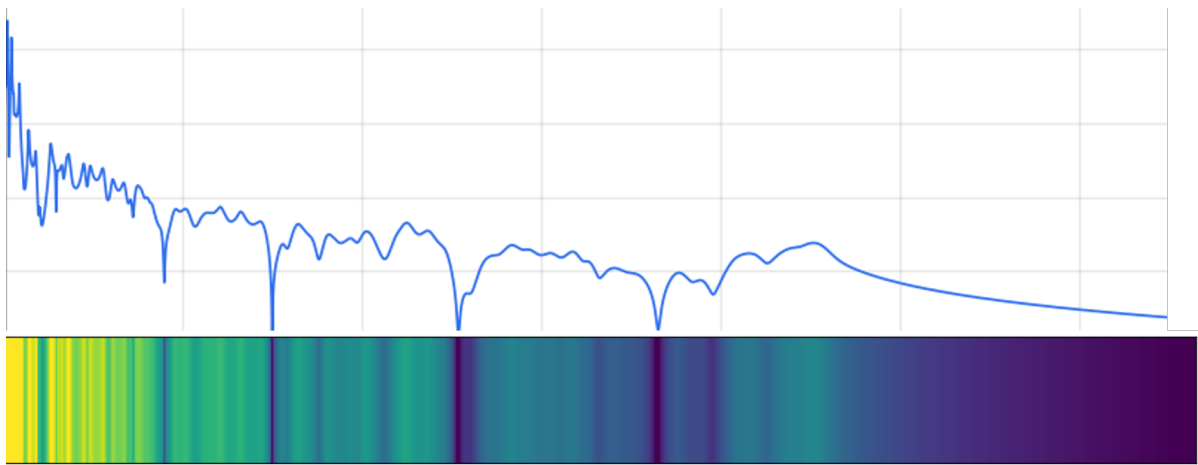


Figure 4.10: Representation of magnitude plot of FRF with equivalent waterfall plot

4.5.5.1. Stitched Waterfall Graph

In this approach, the waterfall graphs from all four sensors are stitched end to end to form a single row of input values. This input is then given to a simple 1D CNN with 3 convolution blocks with 32, 64 and 128 being the output channels with 31, 15 and 7 being the kernel sizes. The convolution block has ReLU, batch normalization and dropout included in it.

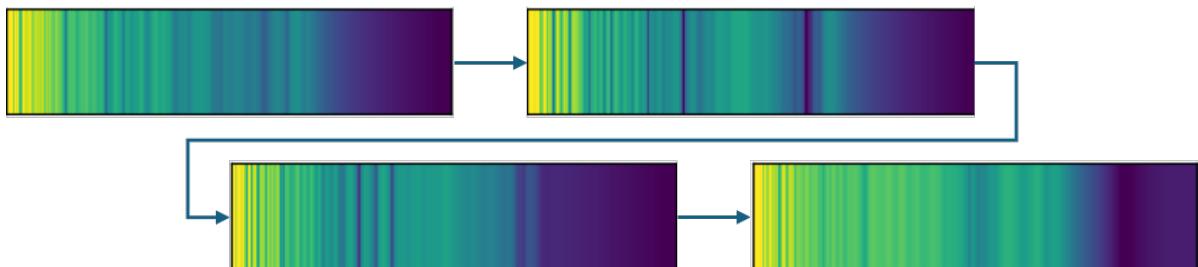


Figure 4.11: Representation of stitched waterfall image used as input for simple 1D CNN

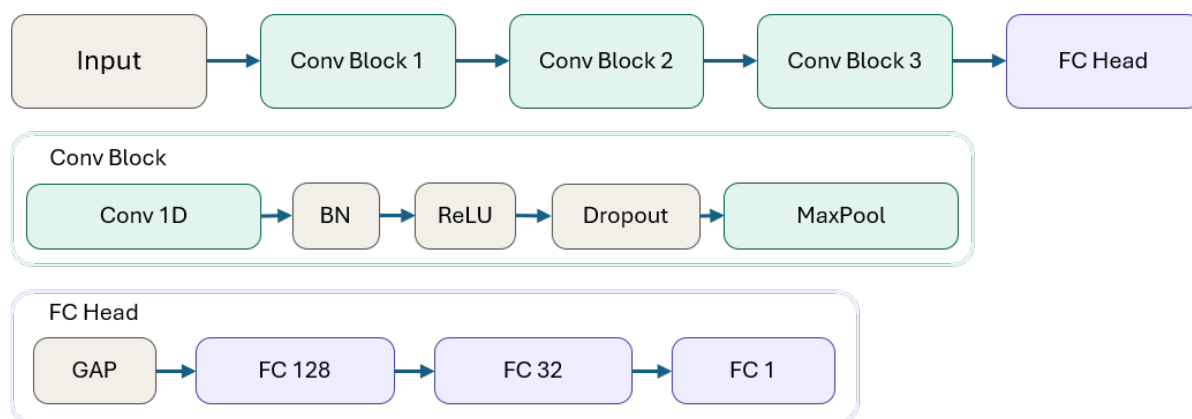


Figure 4.12: Basic network architecture of 1D CNN
BN - Batch Normalization, GAP - Global average pooling

4.5.5.2. Stacked Waterfall Graph

In this approach, the waterfall graphs of the four sensors are stacked on top of each other to form a sort of image. This input is given now to a simple 2D CNN again, with the same number and size of convolution blocks, with the only difference between them being that the kernel now sweeps across all 4 sensors.

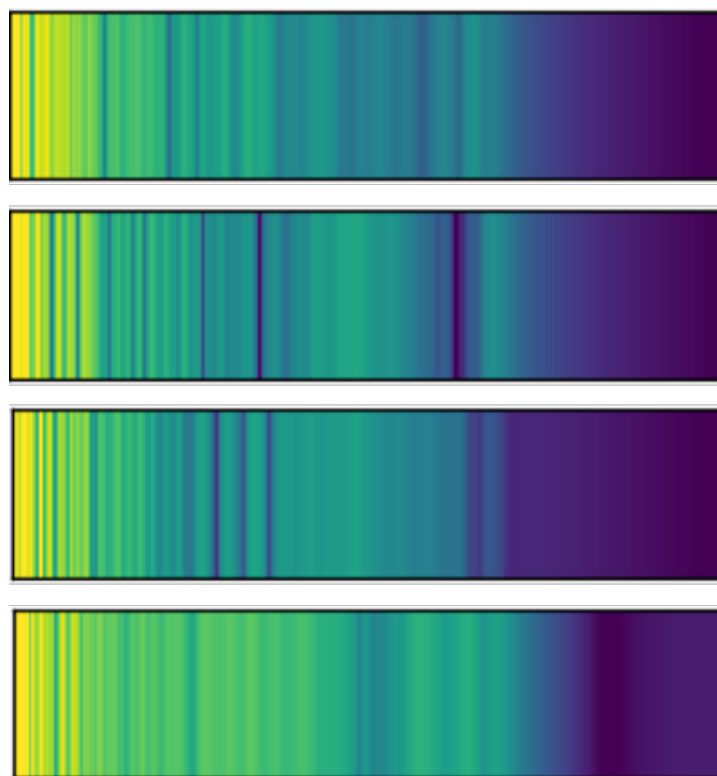


Figure 4.13: Representation of stacked waterfall image used as input for simple 2D CNN

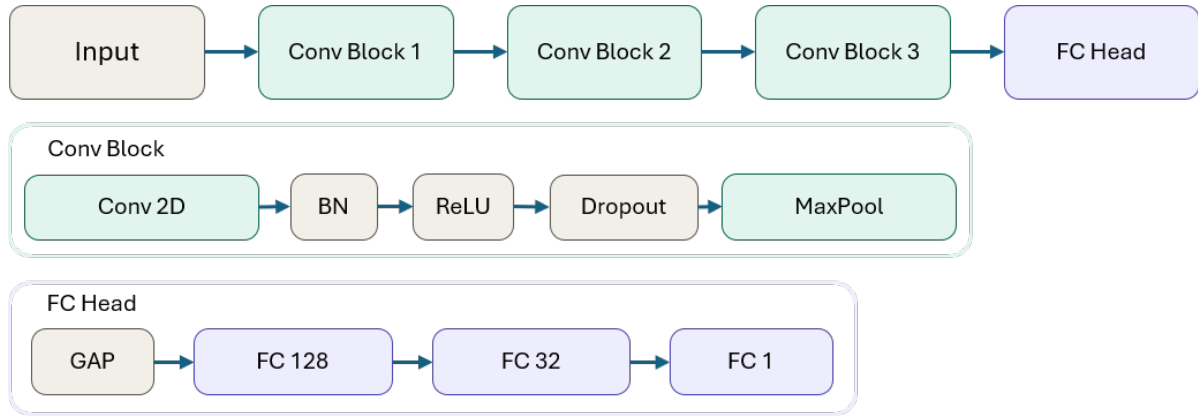


Figure 4.14: Basic network architecture of 2D CNN
BN - Batch Normalization, GAP - Global average pooling

4.5.6. Neural Networks for damage detection on smaller plate

Apart from using the above mentioned networks for detecting damage on the larger plate, some of them were also tested on the smaller plate. This study was limited to only two damage configurations namely Config_1 and Config_4, with only the CNN + MLP, CNN + KAN and 1D CNN with stitched waterfall graph being used. The main aim of this investigation is to see whether the networks perform the same or better when tasked with detecting the same damage in a smaller plate, where the damage might be more apparent. A similar sparse sensor setup with four sensors and 300 training samples was used for this plate to maintain consistency.

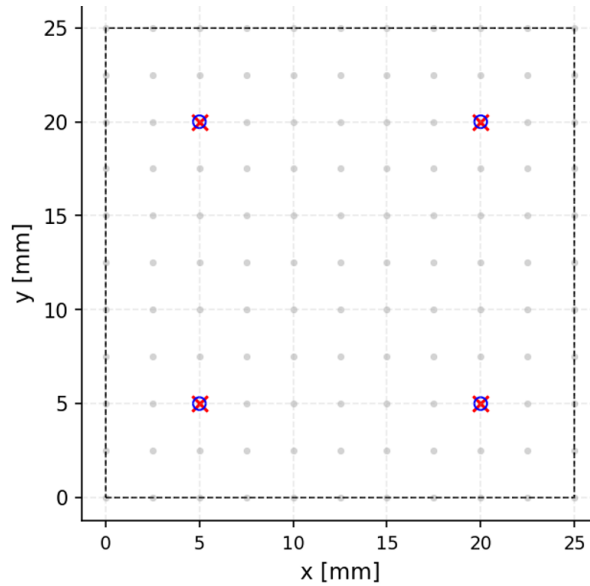


Figure 4.15: Location of sensors for the small plate used for neural network training

4.5.7. Analysis metrics

To evaluate the performance of the various neural networks, their sensitivity, specificity and balanced accuracy will be compared. These metrics are chosen as they represent the model's ability to pick out damaged and healthy samples. Here, damaged samples are the positive class, while healthy samples are the negative class.

Sensitivity is the ability of the model to correctly detect damaged samples. It is defined as the fraction of damaged samples which are correctly classified.

$$Sensitivity = \frac{TP}{TP + FN} \quad (4.5)$$

Specificity is the ability of the model to correctly detect healthy samples. It is defined as the fraction of healthy samples which were correctly identified.

$$Specificity = \frac{TN}{TN + FP} \quad (4.6)$$

Balanced accuracy is the combined measure of the model's performance on both healthy and damaged sample classification. It is simply the average of the specificity and sensitivity.

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2} \quad (4.7)$$

In all these metrics, True Negative (TN) means the number of samples correctly classified as healthy, False Negative (FN) means the number of samples incorrectly classified as healthy, True Positive (TP) means the number of samples correctly classified as damaged, and False Positive (FP) means the number of samples incorrectly classified as damaged.

5

Results & Analysis

The following chapter discusses the results obtained from both the population and non-population based methodologies described previously. The analysis focuses on the ability of the different approaches to detect small/micro-scale damages.

For the population based approaches, the effects of using different architectures, as well as data representations combined with different damage sizes and severities for damage detection, are examined. For the non-population approach, the effects of different mesh sizes combined with different damage sizes and severities are examined.

5.1. Neural Network Results

To examine the performance of the different architectures, some common metrics have been established previously, namely, sensitivity, specificity and balanced accuracy. In each of the cases, performance metrics across the different datasets are shown to figure out trends in performance.

5.1.1. Results - CNN + MLP

From studying the metrics in the four plots below, two common trends can be seen. The first trend is with respect to the size of the damage. This trend is more apparent, and clearly, it can be seen that in this network, as the damage size increases, the network's performance increases by nearly 20 points in all the metrics compared to the case with the smallest size and least severity.

The second trend, which can be observed, is the influence of damage severity. This trend is much less pronounced than the first trend and more evident in the smaller sized datasets, where there is a roughly 5 point improvement.

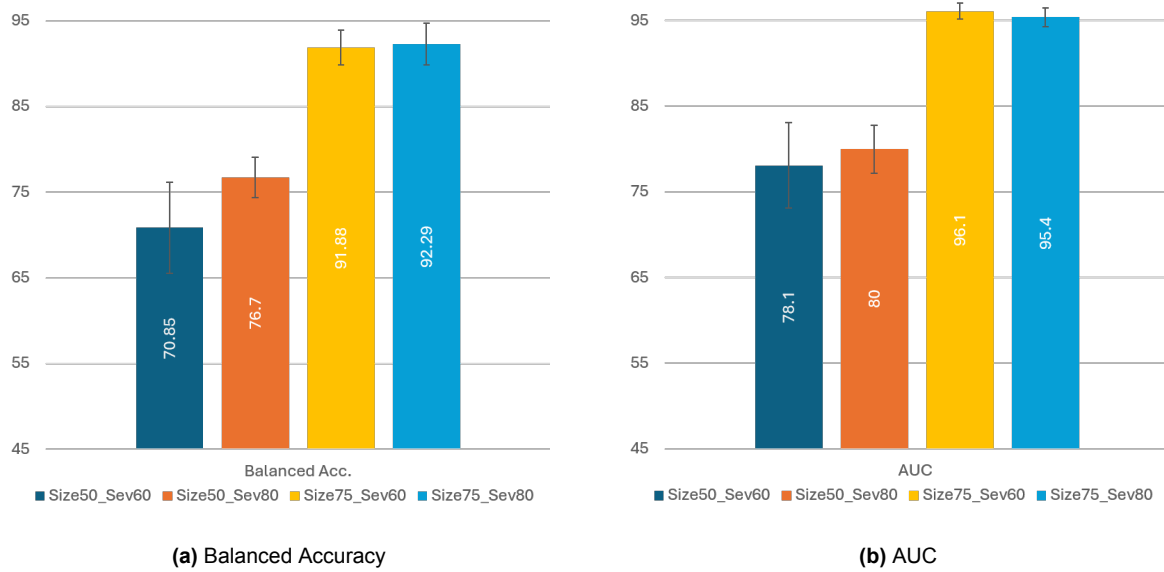


Figure 5.1: Balanced accuracy and AUC trends for the CNN + MLP network
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

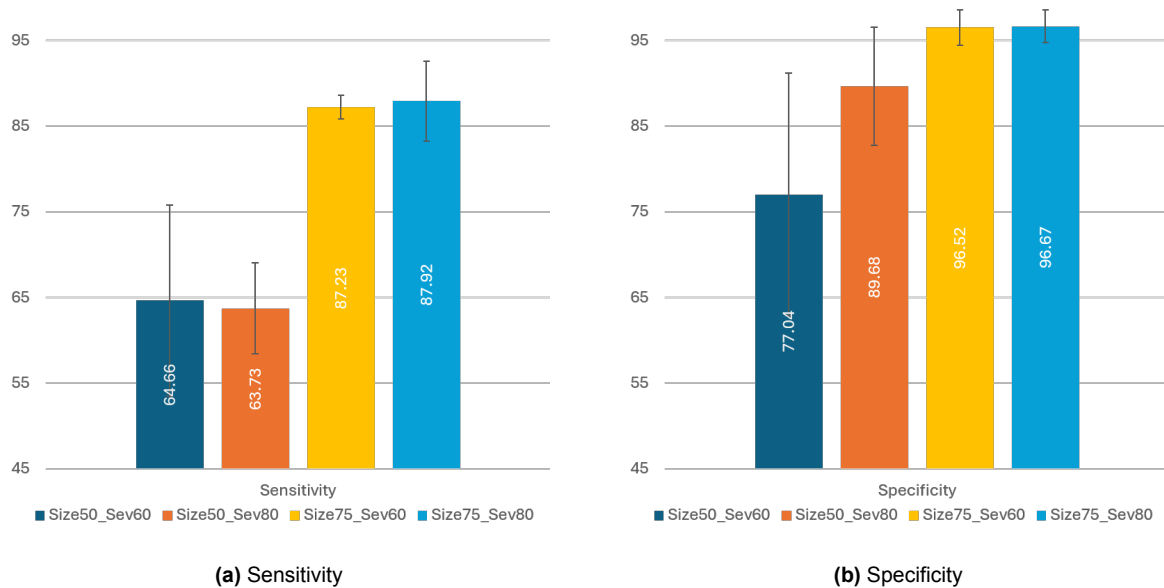
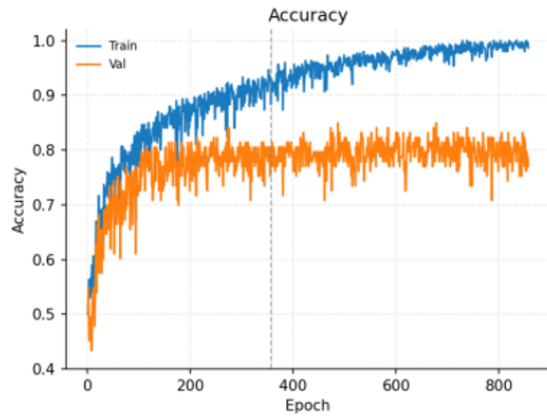
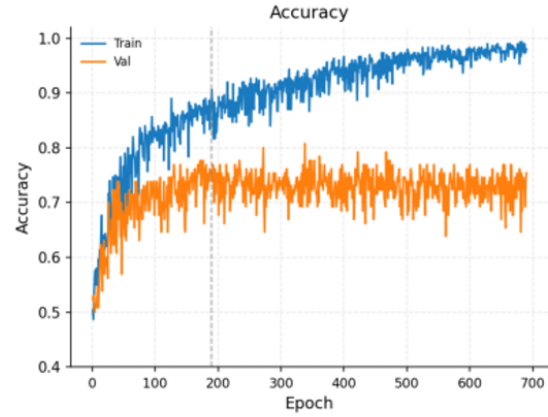


Figure 5.2: Sensitivity and specificity trends for the CNN + MLP network
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

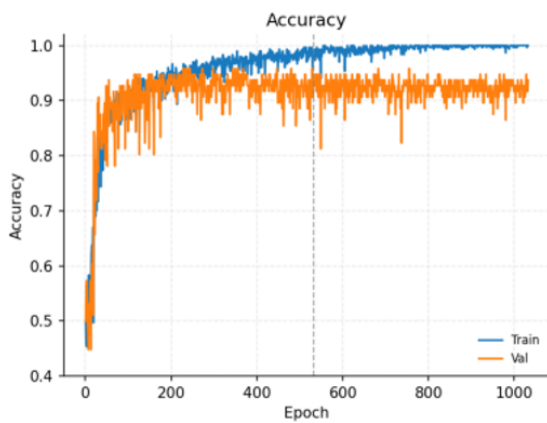
The other aspect to keep in mind is the performance on the training set versus the validation set to evaluate the amount of overfitting. From the plots below, we can see that at the smaller size, there is a lot of over fitting, compared to the larger sizes, while with varying severity, the over fitting seems to remain about the same.



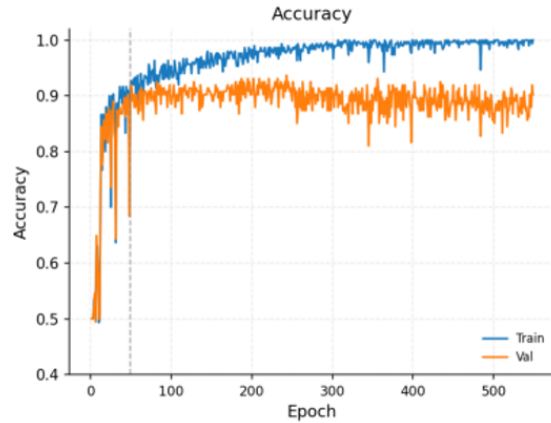
(a) Accuracy - Config_1



(b) Accuracy - Config_2

Figure 5.3: Accuracy plot for the CNN + MLP network to show overfitting trends

(a) Accuracy - Config_3



(b) Accuracy - Config_4

Figure 5.4: Accuracy plot for the CNN + MLP network to show overfitting trends

5.1.2. Results - CNN + KAN

In this architecture as well we can see that the general trend observed in the previous network continues but to differing extents. Where in the CNN + MLP network the smaller sized datasets, the balanced accuracy stayed between 70 to 80%, here we can see that even for the smallest damage size with the lowest severity, balanced accuracy stays above 80%.

There is an anomaly noted here with the Size75_Sev60 dataset where the values seem to be breaking the trend, but this can be attributed to one of runs erroring out thus pulling the entire mean down. More likely than not, in this run the specificity was very high with a sensitivity close to 50%.

The drastic improvement here could possibly be attributed to the addition of the KAN as the head instead of the older MLP head; perhaps its ability to better learn the nonlinear relations bolstered its performance.

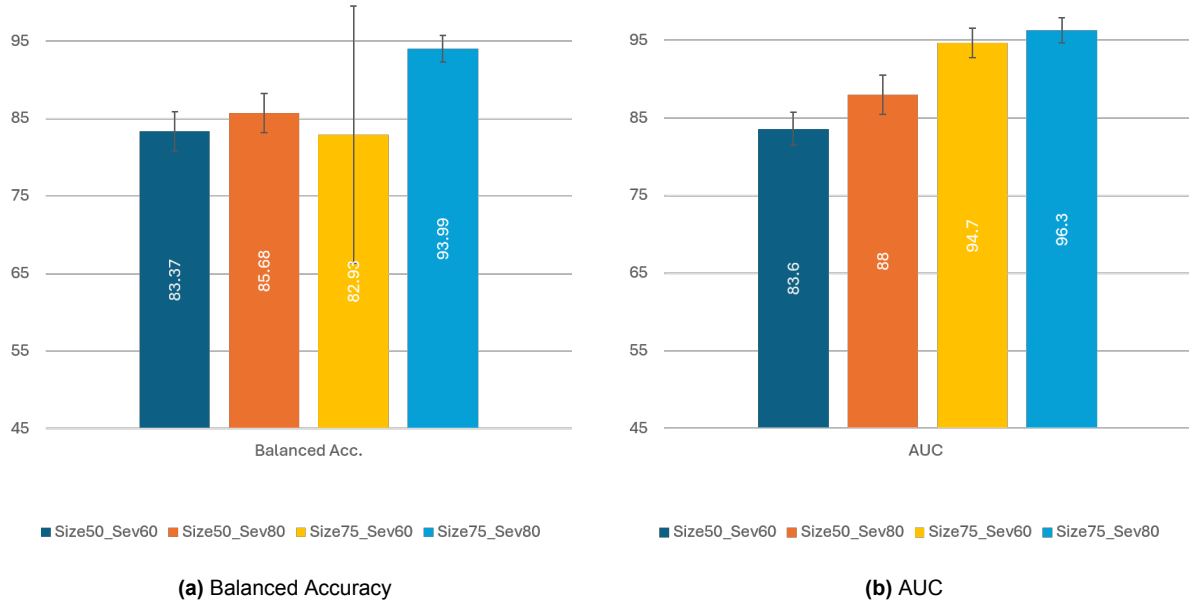


Figure 5.5: Balanced accuracy and AUC trends for the CNN + KAN network
 Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

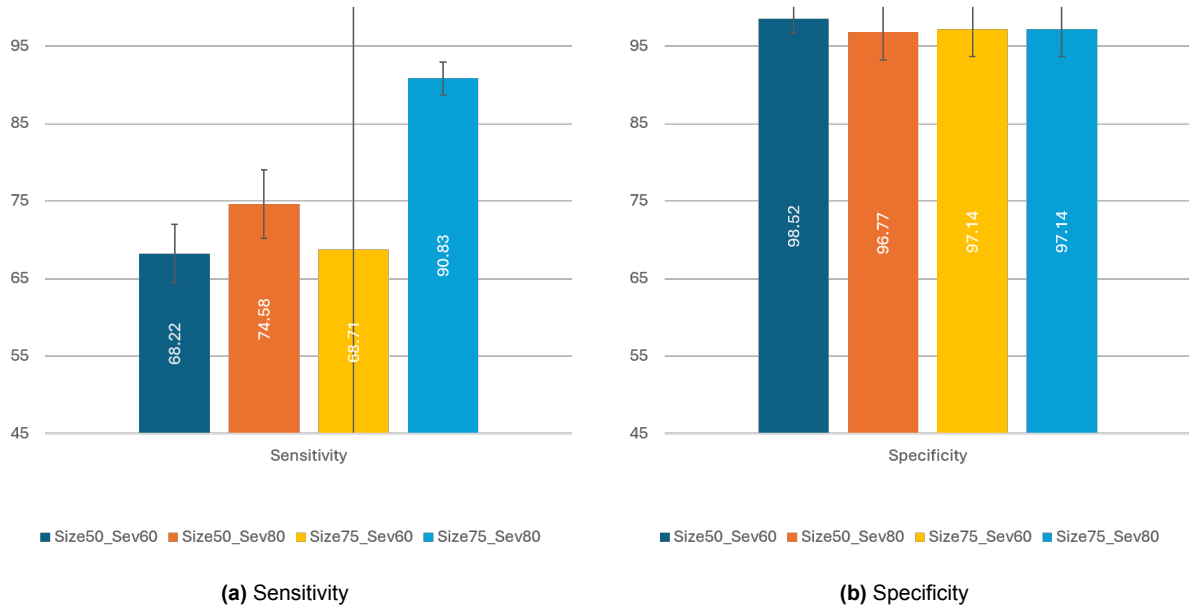
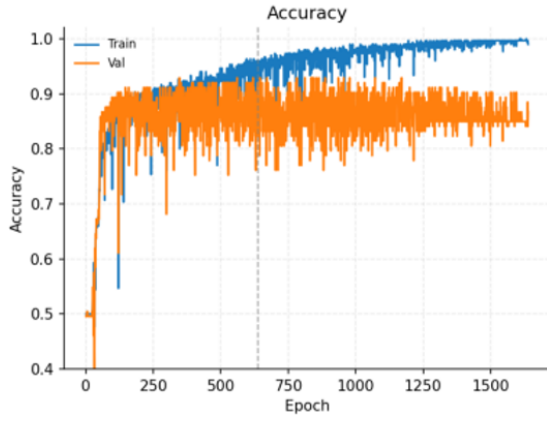
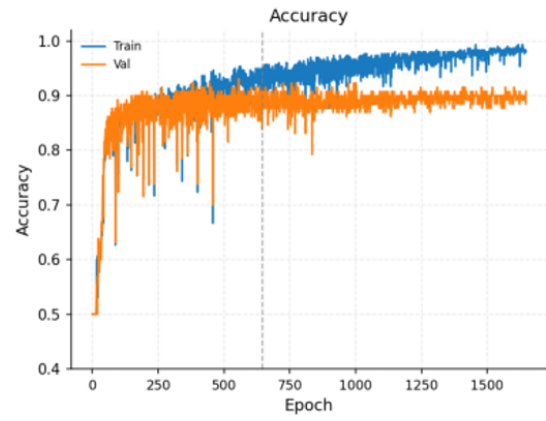


Figure 5.6: Sensitivity and specificity trends for the CNN + KAN network
 Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

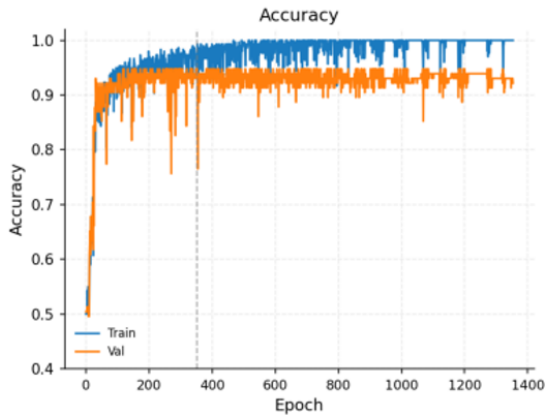
In the accuracy plots for the CNN + KAN network, we can already observe that overfitting is significantly reduced and that the cases with smaller damage size and severity are performing much better than earlier while the other two datasets also show minor improvement. Overall, a similar trend to the CNN + MLP network is seen.



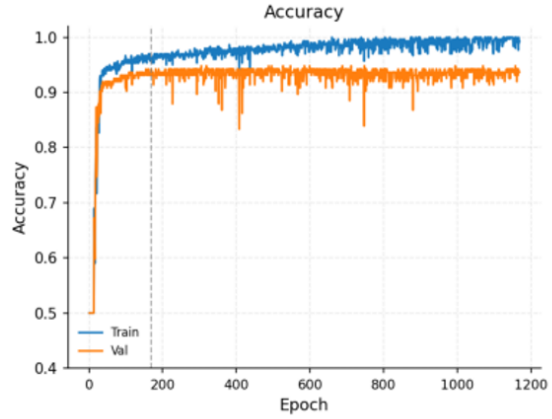
(a) Accuracy - Config_1



(b) Accuracy - Config_2

Figure 5.7: Accuracy plot for the CNN + KAN network to show overfitting trends

(a) Accuracy - Config_3



(b) Accuracy - Config_4

Figure 5.8: Accuracy plot for the CNN + KAN network to show overfitting trends

5.1.3. Results - KAN

With all networks till now being relatively shallow, the KAN approach switches things up with a deep 8 layer network. Here, only one of the two trend is visible, namely the improvement with increase in size. The performance jumps about 20 points in balanced accuracy with near perfect accuracy. Like in the previous CNN + KAN case, we can see that there is an anomaly in the Size50_Sev80 dataset, where once again one of the repetitions has errored out.

In regards to overfitting, all KAN cases display the same trend of improvement across both damage size and damage severity as in the previous networks. Overfitting is lower than in the preceding networks but with the tradeoff of requiring to be trained for more epochs.

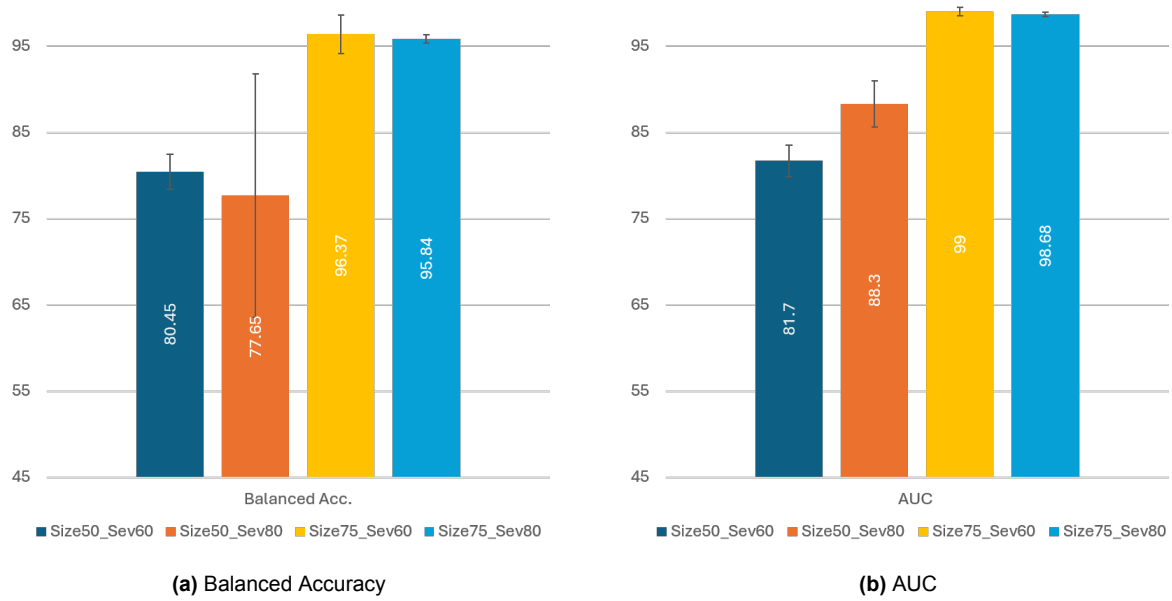


Figure 5.9: Balanced accuracy and AUC trends for the KAN networks
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

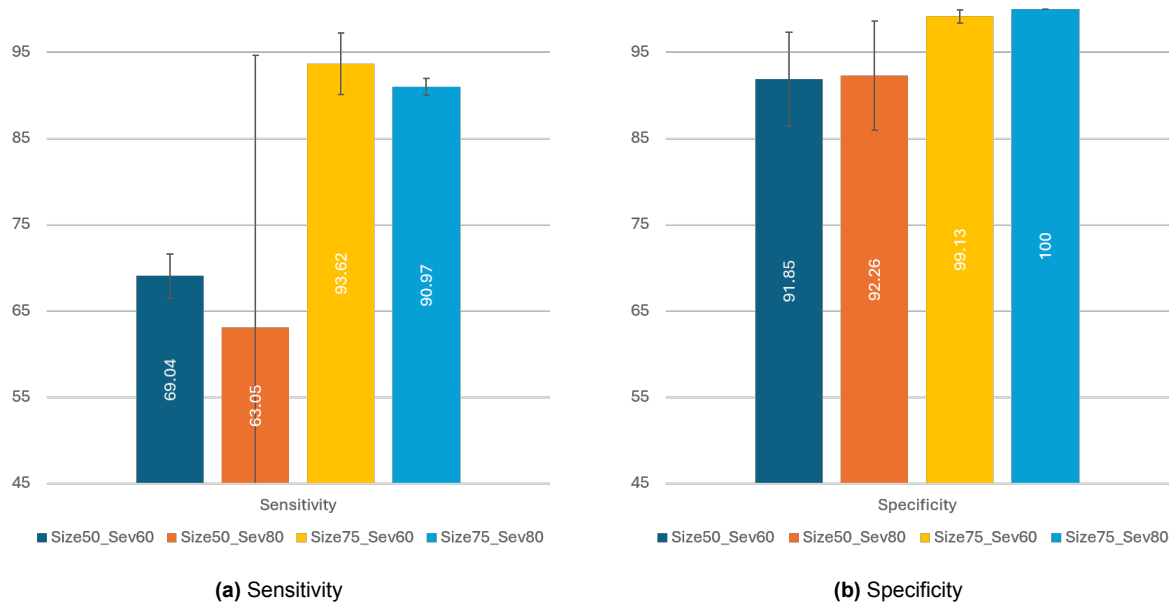


Figure 5.10: Sensitivity and specificity trends for the KAN network
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

5.1.4. Results - CNN (Waterfall Graph)

The following sections shows the results of using a simple 1D and 2D CNN coupled with different waterfall graph based inputs for the purpose of damage detection. Waterfall graphs were used as a method to check if slight modifications to the standard FRF waveform could help improve performance.

In general, both approaches perform at par or better than any of the previous networks and get trained in a much shorter duration. Of the two approaches, the stacked approach per-

forms better even on the dataset with smallest size and severity, surpassing all the previous networks.

As in the previous networks, the major differentiating factor is the performance of the network on the datasets with the smaller damage size, with performance on the larger damage size datasets being nearly at par in all implementations

5.1.4.1. Stitched Waterfall Graph

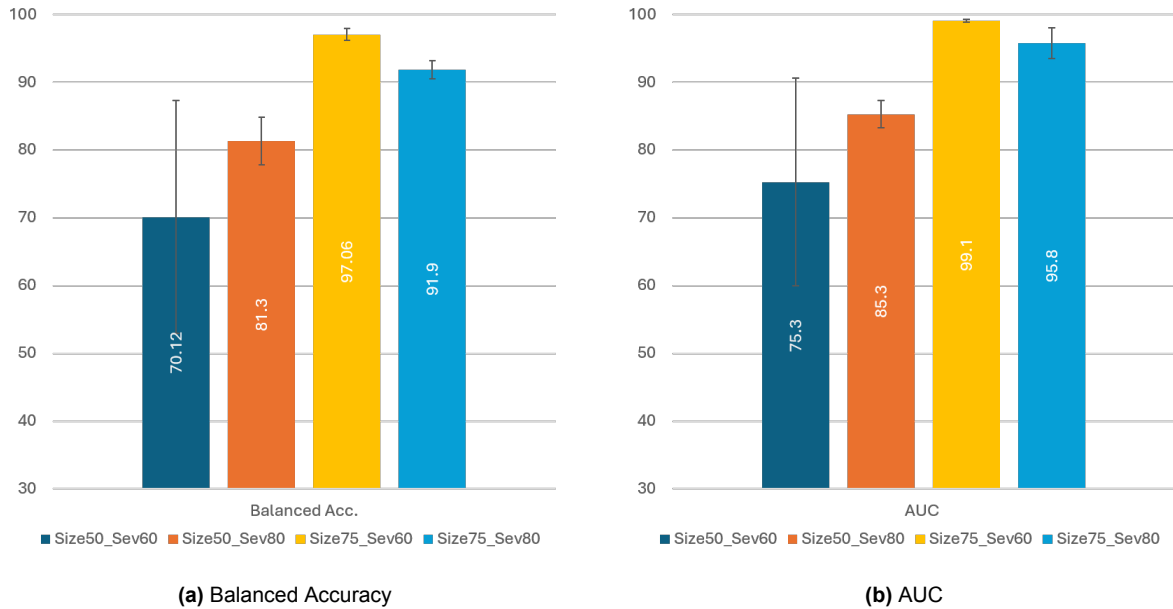


Figure 5.11: Balanced accuracy and AUC trends for the 1D CNN network with waterfall graph
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

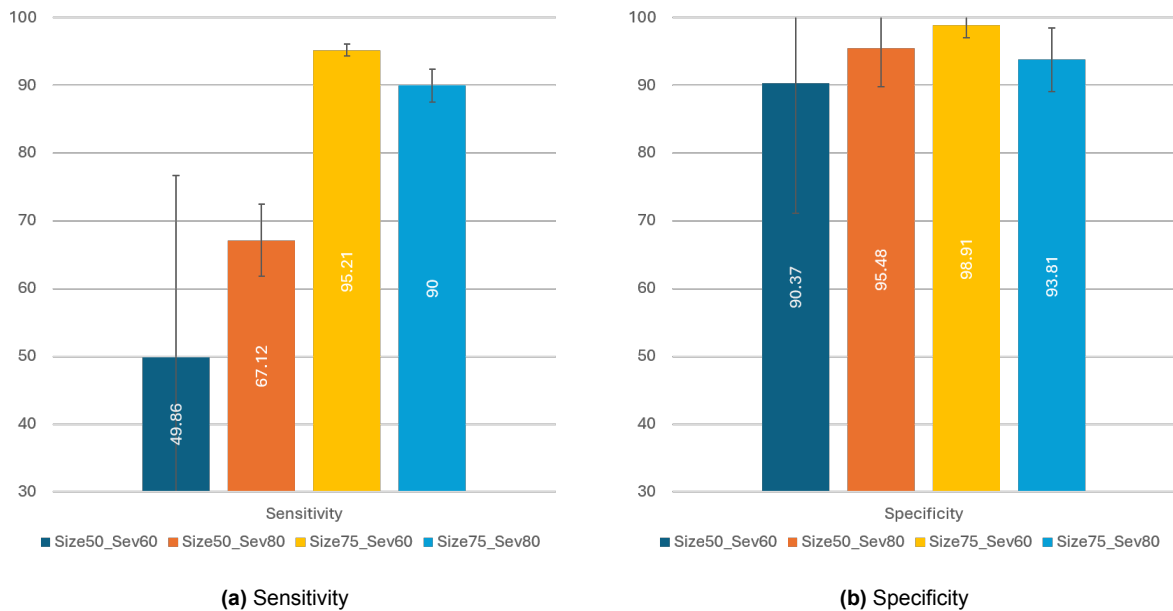


Figure 5.12: Sensitivity and specificity trends for the 1D CNN network with waterfall graph
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

5.1.4.2. Stacked Waterfall Graph

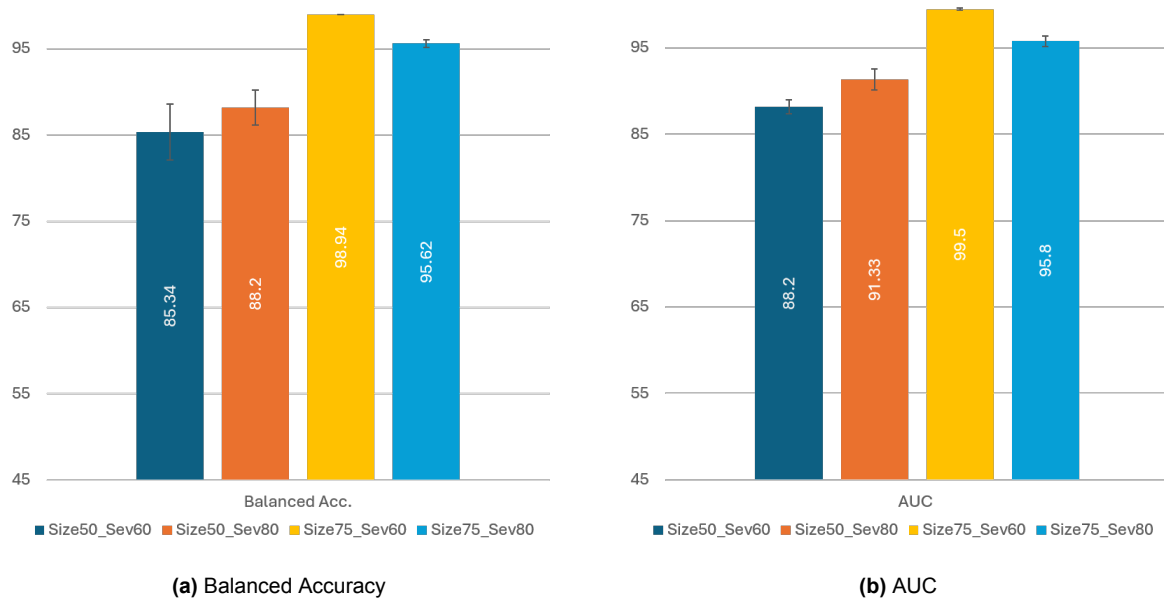


Figure 5.13: Balanced accuracy and AUC trends for the 2D CNN network with waterfall graph
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

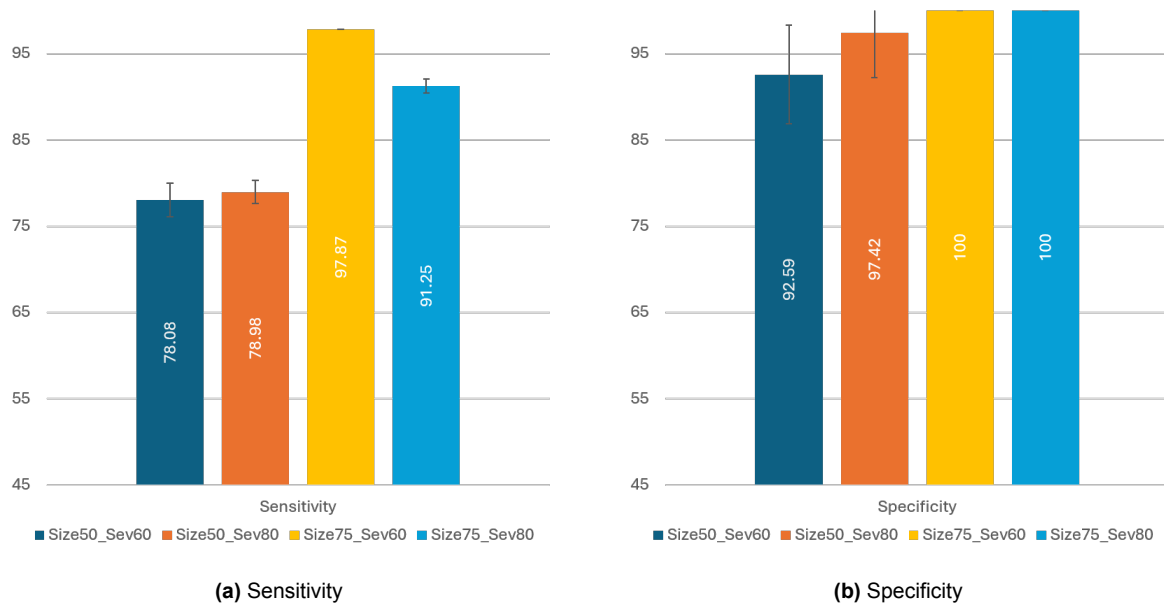


Figure 5.14: Sensitivity and specificity trends for the 2D CNN network with waterfall graph
Config_1 - Dark Blue, Config_2 - Orange, Config_3 - Yellow, Config_4 - Light Blue

5.1.4.3. Learnings from SHAP

As discussed in the methodology, interpretability and explainability are increasingly important to properly understand what the network learns during training and what it concentrates on. This information can significantly help streamline not only the network but also the data. With this in mind, SHAP was applied to the CNNs training on the waterfall graph based data, from

which a certain region of the data was repeatedly seen as important. The FRF data is stored as 6000 frequency bins covering the entire band from 10 Hz to 1 MHz, with each bin roughly averaging over 160 - 170 frequencies.

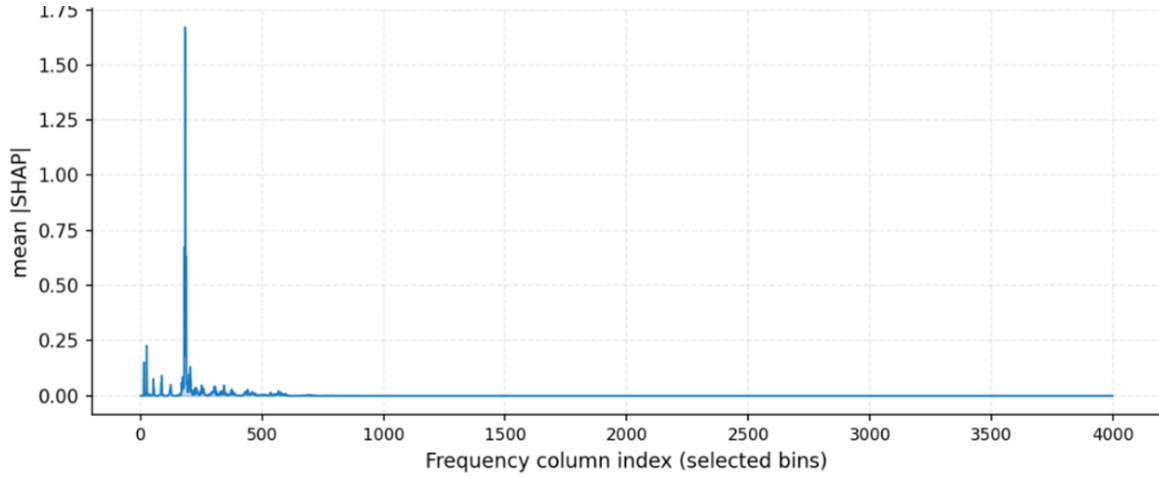


Figure 5.15: SHAP importance plot for 2D CNN network on Config_4

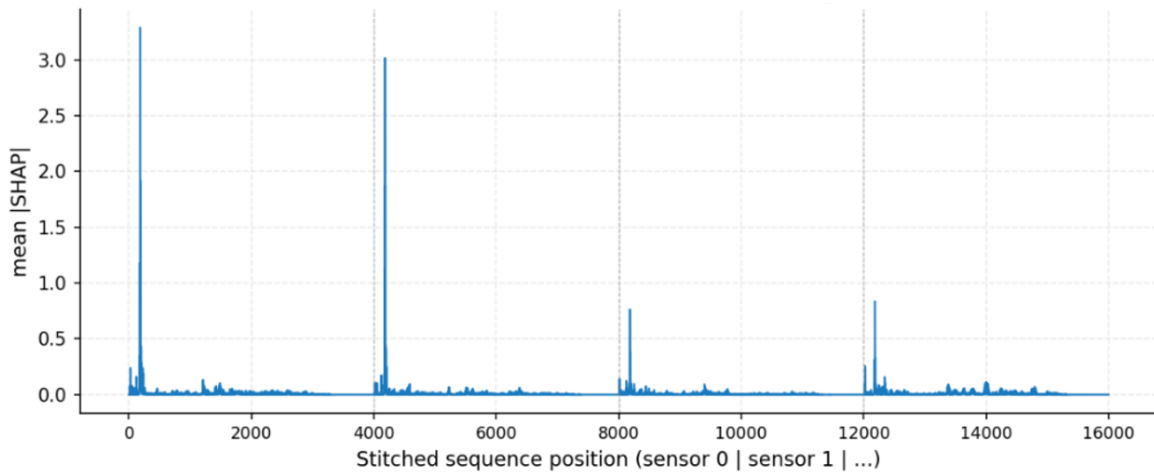


Figure 5.16: SHAP importance plot for 1D CNN network on Config_4

In the two above plots, it can clearly be seen that both the networks favour a particular region in the beginning of the input, roughly lying between the first 500 to 1000 frequency bins, with lesser importance given to some of the later bins. A major peak is seen around 30 to 40 kHz, with the rest extending roughly up to a frequency of 200 kHz, which could mean one of two things: first, that there is no meaningful information beyond this range for the network to learn from or that learnable data should exist in the region, but due to the size of the mesh chosen, only the range up to 200 kHz could be resolved accurately and that finer meshes are needed for more accurate solutions.

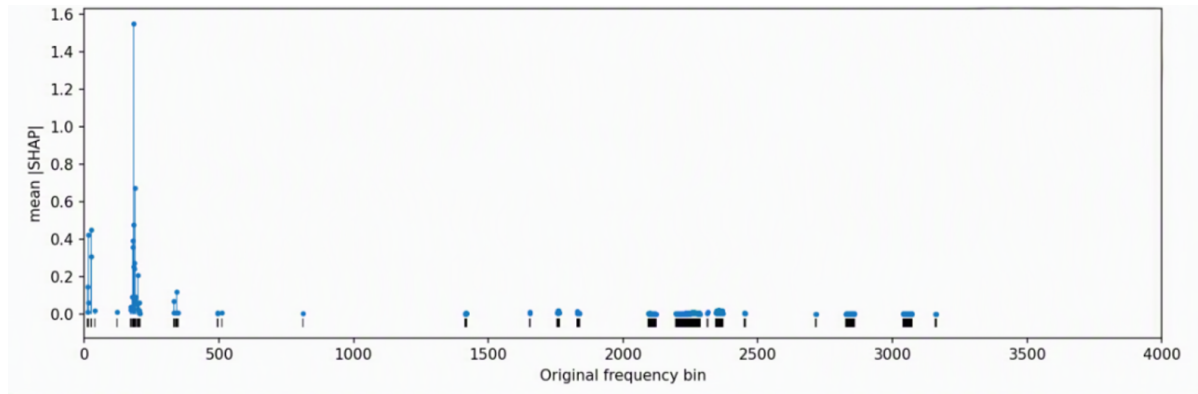


Figure 5.17: SHAP importance plot for CNN + MLP network on Config_4 selected through LARS

The same SHAP was applied to the other networks, learning from the feature selected input, which also shows a similar pattern to the ones using the waterfall graph. More importance is given to a region up to 1000 bins (roughly 200 kHz), with an especially high importance to bins around 30 to 40 kHz. The rest of the spread is pretty similar to the previous ones as well, with multiple bins in the higher end being chosen but with lower importance.

This shows that, irrespective of the way the data is formatted or pre-processed, the network picks the same/similar bins with nearly the same importance as well. The next step would be to either improve the mesh resolution to capture the information more accurately or to reduce the upper end of the band to a more manageable level, such that a lesser number of frequencies can be averaged into each bin, making the FRF more accurate.

5.1.5. Neural Network Result from smaller plate

An additional investigation was made into how the CNN based networks would perform when used on a smaller plate. This plate was essentially one quarter of the coupon sized plate, with its dimensions being 25 mm x 10 mm x 4 mm. The plate had a mesh of 10x10x24.

The smaller plate was also subject to the same training parameters as the bigger plate, as well as the number, size and severity of damage. The study was limited to two datasets, one with smallest damage size and severity (Config_1), and the other with largest damage size and severity (Config_4). The following are the results of said investigation.

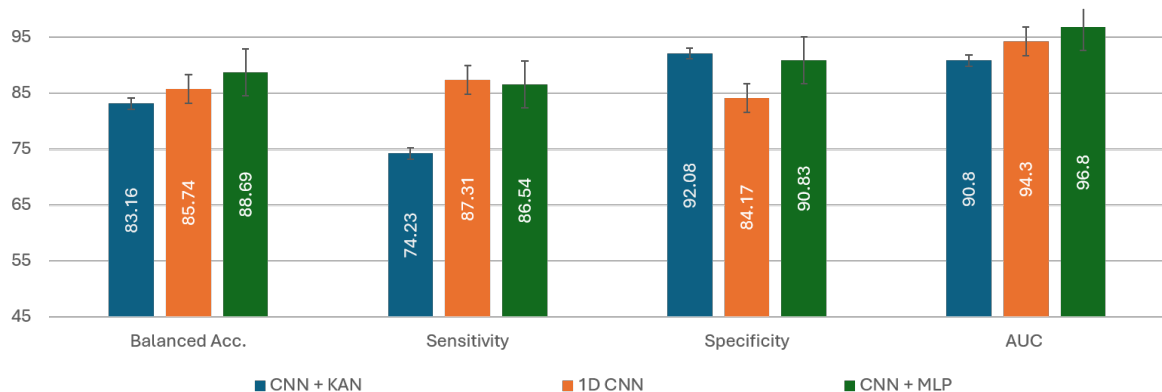


Figure 5.18: Performance figures of different networks on the small plate dataset with damage size 0.5mm and severity 60 (Config_1)

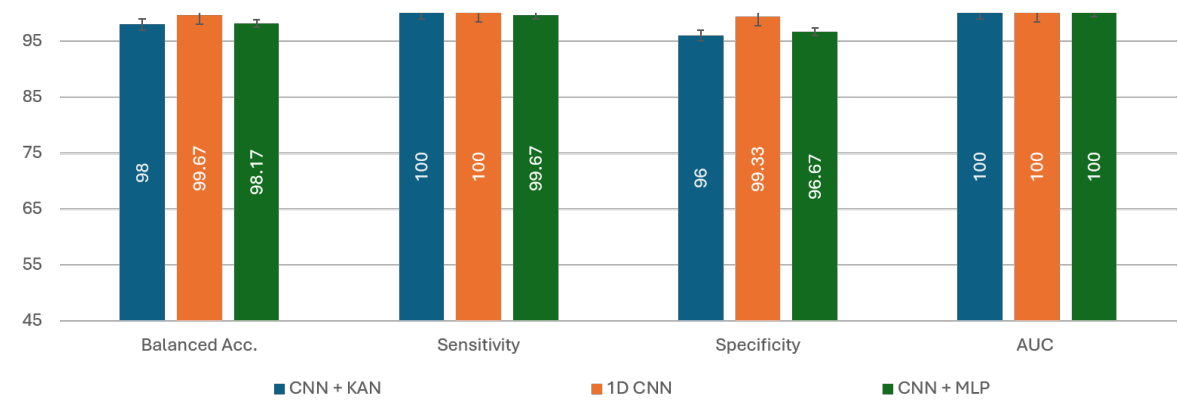


Figure 5.19: Performance figures of different networks on the small plate dataset with damage size 0.75mm and severity 80 (Config_4)

From both plots, it can be seen that performance on the small plate is better overall compared to the larger plate for the same network, with all three networks performing nearly the same in the second case. This can perhaps be attributed to the fact that the ratio between damage size and plate size has increased, which could make the damage more evident, thus in turn improving detection.

5.2. ULS Results

This section deals with the results of the non-population based approach. the results presented are for the various configurations of mesh size, damage size and damage severity with common damage locations serving as the comparison points.

Surface Mesh	Damage Size (mm)	Damage Severity (%)	Damage Count
10×5 elements (Mesh 1)	0.5	60	0 to 5 damages
		80	
	0.75	60	
		80	
20×10 elements (Mesh 2)	0.5	60	0 to 5 damages
		80	
	0.75	60	
		80	
30×15 elements (Mesh 3)	0.5	60	0 to 5 damages
		80	
	0.75	60	
		80	

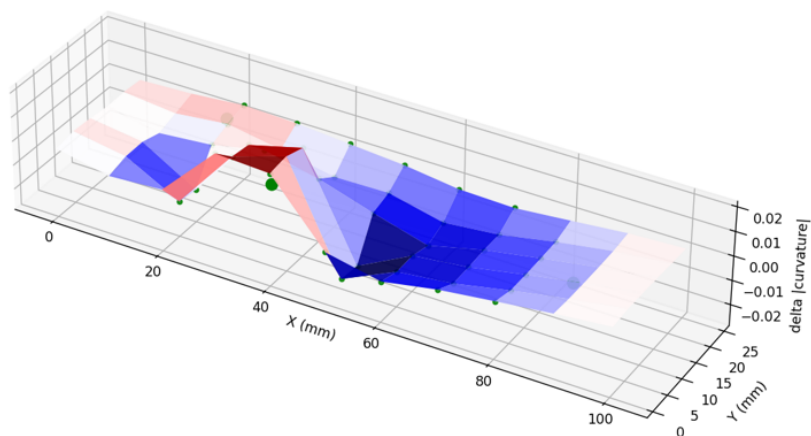
Table 5.1: Damage configuration for non-population approach

5.2.1. Effect of Damage Size

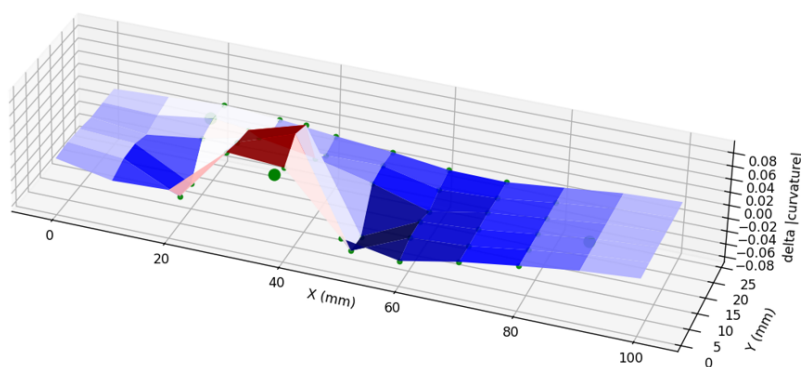
The first effect being studied is the effect of damage size on the visualization capability of the ULS plot, in this case only the ULS curvature difference indicator is used as it is found adequate for the current purpose. To show the differences, two different mesh sizes are chosen with a fixed damage severity. For this visualization, a damage case with 3 damage sites is chosen so that effect of size can be studied at multiple locations.

From the below plots, comparing a,b and c,d we can see that keeping the mesh and damage severity constant, the size of the damage does impact the visibility of the damage even if only to a small margin. In both cases, the shapes of the peaks slightly change while the major change is in the amplitude of the peak, which in both cases is about 4 time higher.

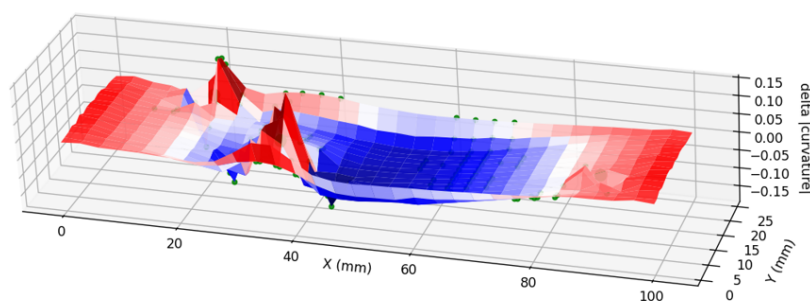
In the cases with Mesh 1, only one of the damages is visible as a large prominent peak with the other damages not being visible. In the case of Mesh 3, Two larger peaks can be seen on the left side of the plate with a small bump being visible near the right end.



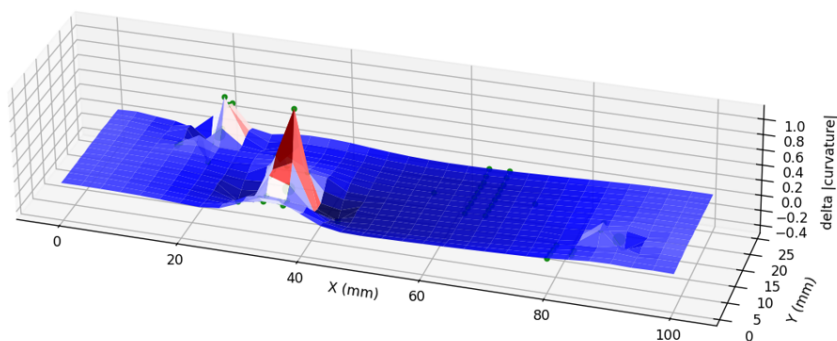
(a) Mesh 1 - Size 0.5 mm, Severity 80



(b) Mesh 1 - Size 0.75 mm, Severity 80



(c) Mesh 3 - Size 0.5 mm, Severity 80



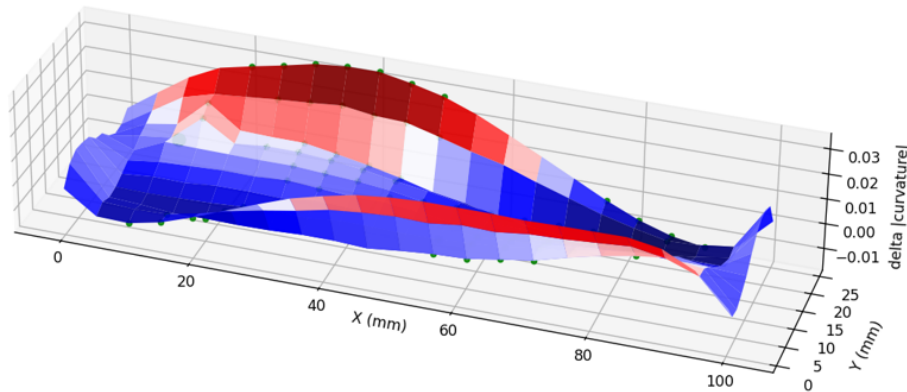
(d) Mesh 3 - Size 0.75 mm, Severity 80

Figure 5.20: ULS plots of Mesh 1 (a,b) and Mesh 3 (c,d) with three damage points, showing the impact of damage size.

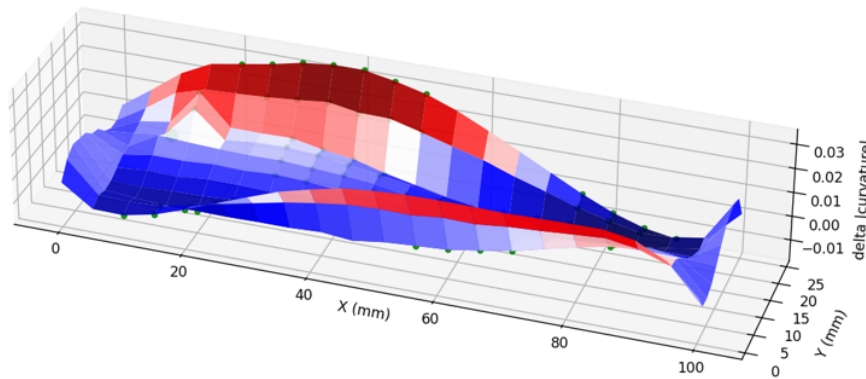
5.2.2. Effect of Damage Severity

The next effect being studied is the effect of damage severity on the ease of visualization. Two mesh sizes are compared to see how well the effect carries over, in this case a damage case with a single point of damage is chosen to see how big of an effect severity has.

In image a and b of 5.21, a small peak can be seen in the back corner of the plate while the peak is not very prominent in either plot, a closer inspection shows that the peak in the case with higher severity is comparatively larger, even if the overall plot lies in the same value range. In 5.22, it is more clearly visible that the higher severity case has a higher peak thus making locating the damage much easier even in a case with only a single damage point.

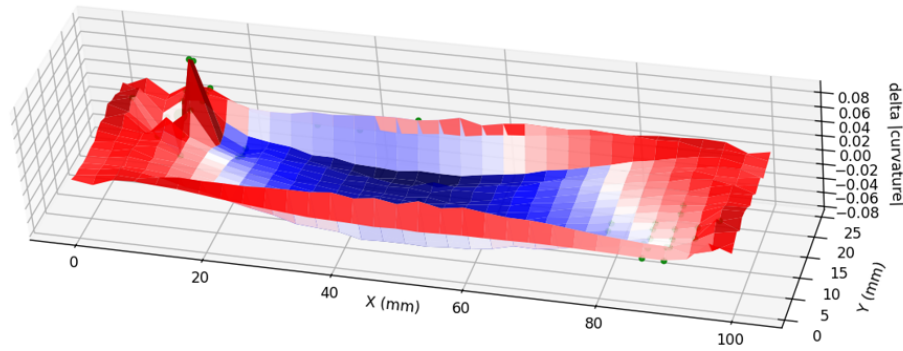


(a) Mesh 2 - Size 0.75 mm, Severity 60

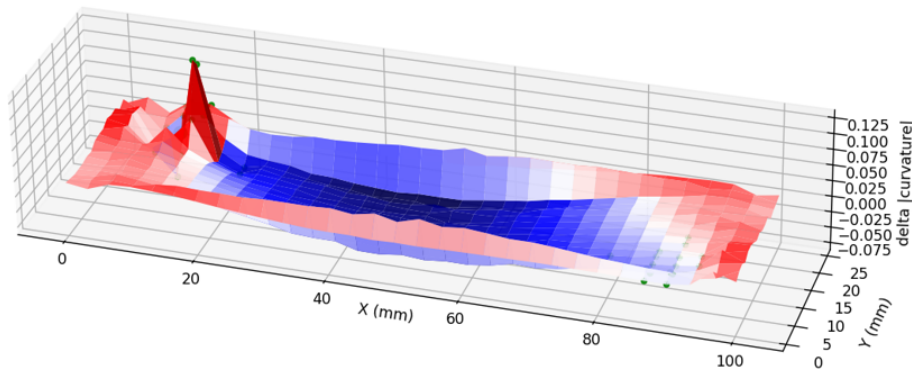


(b) Mesh 2 - Size 0.75 mm, Severity 80

Figure 5.21: ULS plots of Mesh 2 with one damage point, showing the impact of damage severity.



(a) Mesh 3 - Size 0.75 mm, Severity 60



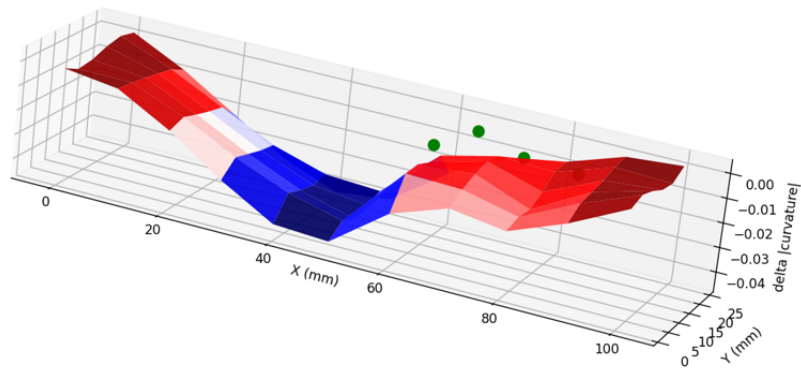
(b) Mesh 3 - Size 0.75 mm, Severity 80

Figure 5.22: ULS plots of Mesh 3 with one damage point, showing the impact of damage severity.

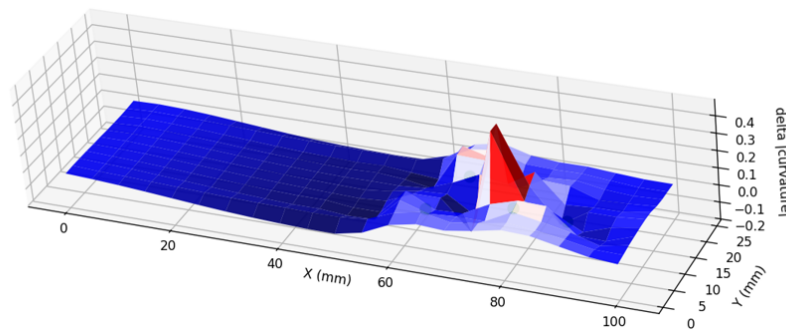
5.2.3. Effect of Mesh Size

The final effect being studied is the effect of mesh size on the ease of locating the damage in the visualization. For this, the damage size and severity have been fixed at 0.75mm and 80 respectively, based on which plots for the three different mesh setups are generated. The damage case chosen is one with five damage location clustered towards the right end of the plate.

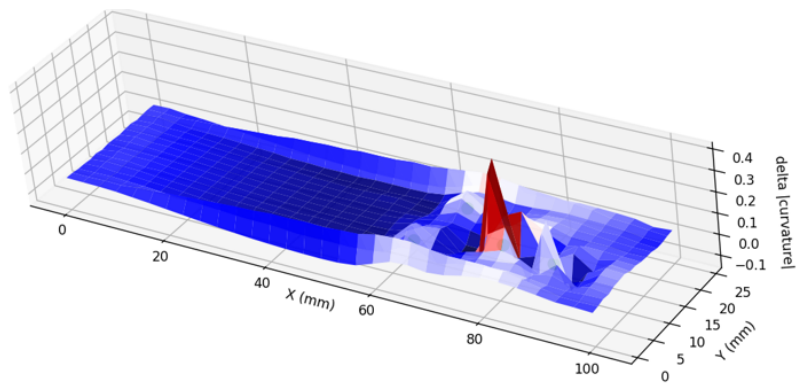
It can clearly be seen that as the mesh size becomes finer, the damage peaks are visible much more easily. In Mesh 1, the cluster shows up as a single large bump with no information regarding how many damage sites might be present, but in Mesh 2, there is more structure to the peaks with a central larger peak and a smaller peak visible behind it. In Mesh 3, more peaks are visible with two to the left of the central peak and one more to its right



(a) Mesh 1 - Size 0.75 mm, Severity 80



(b) Mesh 2 - Size 0.75 mm, Severity 80



(c) Mesh 3 - Size 0.75 mm, Severity 80

Figure 5.23: ULS plots of Mesh 1, Mesh 2 and Mesh 3 with three damage points, showing the impact of mesh size.

Part III

Conclusions and Future Work

6

Conclusion

This chapter provides a summary of all the findings in this work and addresses each research question and sub-question. The main aim of the study was to find out whether small/micro-scale damage can be detected in metals using both population and non-population based methods.

This work was undertaken by looking at the increasing complexity of structures and components, especially in the aerospace industry, coupled with more advanced and faster manufacturing techniques, which place a greater demand on structural health monitoring. Conventional methods might require parts to be taken out of service, leading to higher downtime and cost, for an industry that demands ever increasing efficiency, an inspection method that can easily be integrated into the lifecycle is a major boost.

The work in this thesis is a first step towards the goal of rapid, in situ SHM/NDDD. It investigated how vibration signals could be used to detect small/micro-scale damage in coupon sized metal plates using both neural networks with sparse sensing as well as vibration based damage indicators. Multiple datasets were generated by varying damage size and severity to assess how the networks performed across different ranges and to see if some common trends could be seen and if any one particular architecture performs the best.

Many different network structures were tried, including simple CNNs, modified CNNs, KANs and even amalgamations of these networks. More or less common trends were observed in all networks pertaining to damage size and severity, and out of all the networks, the best performing network was identified across all different damage sizes and severities. Additionally, the effects of varying parameters like damage size, severity and mesh size were also studied using the ULS curvature difference indicator to investigate which combination was best at identifying and locating damage.

6.1. Research Question outcomes

The main research question stated at the beginning was to find out how vibration data can be used to detect small/micro-scale damage in metals. Through the various results shown in the previous chapter, it has been seen that both the population and non-population based methods prove successful in this task to different extents. Apart from the main question, the sub questions have also been answered as shown below.

6.1.1. Research Question 1

This question aimed to investigate whether neural networks trained using vibration data in different forms could successfully detect the presence of small/micro-scale damage in metals. Multiple different architectures/combinations of architectures were used to check performance across two data representations.

The overall result says that neural networks are very well suited for this task, even after using a sparse sensor setup with only 4 nodes. Two different data representations were checked; the first, the full FRF, on which feature selection is carried out using LARS and the second, a waterfall graph representation of the full FRF. The first representation was checked on a CNN + MLP, CNN + KAN and a KAN network, while the other representation was checked using a simple 1D and 2D CNN.

Of the four datasets tested on, across all networks, the lowest performance was an accuracy of about 70% on the Config_1 using the CNN + MLP network, while the best performance on the same data was about 85% accuracy using the 2D CNN network. Of the networks trained on the selected features, the best overall performance was by the KAN only network, followed closely by the CNN + KAN network and the CNN + MLP being the worst performing. Of the other two 1D and 2D CNN networks, the 2D CNN performed better than the 1D CNN on all datasets.

On comparing the performance of the networks trained on the two different representations, the CNNs trained on the waterfall graph data performed much better compared to their counterparts, also taking into account their simpler structure and overall lower training time needed.

From all the networks, two common trends were observed; first, as the severity of damage increased, in the best case, a small improvement in detection accuracy of roughly 5% was observed and second, as the size of the damage increased, in the same case, the detection accuracy increased significantly by nearly 20% in the best case. The improvement due to severity was overshadowed by the improvement due to size. While this was the general trend, the variation in improvement was different across the board.

In all cases, SHAP was used to understand what frequencies the networks focused on, and in all cases, it was seen that major concentration was on frequencies below 200 kHz with heavy importance to frequencies around 30 to 40 kHz and some small importance to higher frequencies.

Research Question 1		
RQ1	Can a population based approach be used to detect small/micro-scale damage using vibration data?	
	RQ 1.1	How do different network architectures perform at the detection task?
	RQ 1.2	What effect does the formatting of the input data have on the performance?
	RQ 1.3	What factors affect the network's performance?

Table 6.1: Research Question 1

6.1.2. Research Question 2

This question looks to answer whether there is another way to detect damage without neural networks and a large sample size. The various studies into Uniform Load Surface curvature showed that it was very well suited for the purpose of detecting such small damages, with the trade off being that for using a single sample, a lot of data points on that sample are needed.

Through the various datasets that were used to study the ULS based approach, it was seen that size was an important factor, often making the peaks and valleys more prominent, but only to a small degree. Other factors were also tested, the first being damage severity, which, just like the damage size, did improve detectability but only marginally.

The second and most important factor tested was the mesh size, where a drastic improvement could be seen with finer meshes. While in the coarser mesh, many nearby damages could coalesce into a single large bump, finer meshes allowed for more accurate peaks, better pinpointing the location of the damage as well.

Research Question 2		
RQ 2	Can a non population based approach be used to detect small/micro-scale defects using vibration data?	
	RQ 2.1	Which metric is best suited for detecting the damage?
	RQ 2.2	How does the size of damage affect the ease of identifying it?
	RQ 2.3	What other factors affect the ease of detecting the damage?

Table 6.2: Research Question 2

7

Future Work

The sections till this point discussed the various methodologies used, the results obtained and how they answered the research questions. While in the course of this thesis, many different techniques were tried, some succeeded, and many failed; there are still many aspects which have gone unexplored.

This chapter discusses some of the limitations that persist at this point in the thesis, as well as some recommendations about future prospects of related research. The recommendations, while numerous, are in no way an exhaustive list, as newer technologies and innovative approaches can open an even wider sea of opportunities.

7.1. Limitations

- **Simulated Data:** In the course of this thesis, only simulation data is used for the studies, which brings with it many idealizations, especially regarding measurement noise. Additionally, all work is carried out using a single mesh size which is not overly fine, this leads to poor representation of higher frequencies in the response.
- **Network Architectures:** The thesis sticks to only a few architectures, namely CNNs and KANs, but there are effectively a limitless number of architectures/ combinations of architectures which can be used for this purpose. Investigating all possibilities one by one is a monumental task and ideally should be approached in an alternate manner.
- **Sensor Placement:** The thesis has investigated only a single four sensor placement, which may or may not be the most optimized setup.
- **Structural Complexity:** All work in the thesis is done on a coupon sized plate, but to be truly useful in a industrial sense larger and more complex structures need to be tested as well.

Based on these limitations, the following are a few recommendations regarding where and how future work effort can be focused to help in making the imagined final use case a reality

7.2. Future Work Recommendations

The work done in this thesis required making certain decisions to allow the work to be finished in time, which meant that many avenues could not be explored. Below are a few suggestions

about the different directions the next steps can be taken in.

7.2.1. Experimental Validation

The work in this thesis is based only on simulated data, and as such, many of its choices lead to rather ideal outcomes, especially in terms of the vibration data; though some effort has been made to include variations by slightly changing the material properties, noise arising from practical sources has not been included.

Another challenge regarding the experimental validation arises from having to replicate the Free-Free boundary condition used for the thesis, which might prove difficult and as such, a study could be undertaken to look into the performance under different boundary conditions.

7.2.2. Advanced Damage Modelling

Currently, the thesis simplifies damage to local stiffness reduction. While this is a very straightforward way to look at the problem, it does open up the possibility to look into different types of damage modelling, which can better represent different types of damage, like cracks, delaminations, holes, etc.

This could perhaps be extended such that, with data about how different types of damages change the frequency response, the network could be adapted to also undertake damage type classification instead of just damage detection.

With more advanced modelling, the same workflow could also be extended to composites, which are essential for the aerospace industry as well.

7.2.3. Larger Structures

The investigation in this thesis is limited to small, coupon sized plates as a sort of first step into this workflow. The next logical step would be to expand the approach to include larger structures like plates and beams.

When further steps are taken to expand the work, it must be kept in mind that the final goal is to have this approach be adapted to complex structures, which can be studied without being disassembled or even without being taken out of service

7.2.4. Sensor Configuration

This work has dealt with using a single sparse sensor setup, placing each of the four sensors roughly in its own quadrant, but this opens up space for an investigation into sensor placements.

Future work could look at optimizing the number or the positioning of the sensors, and perhaps, instead of just studying the out of plane response, a 3D response capture could also make for an interesting topic.

7.2.5. Damage Localization

In its current form, the neural network approach is limited only to the damage detection part of the levels of SHM, and while the ULS based approach can help in localization, the trade-off for sensor density has to be made.

As such, using the same/ an optimized sensor setup to train the same or separate neural network for localization could be an interesting study. Another interesting avenue is to use the

ULS plot as an input to a neural network to see if it can augment the simple FRF data.

7.2.6. Neural Network Architectures

In this thesis, only a few neural network architectures could be studied owing to an incredibly large number of options to choose from. One interesting way this could be tackled is by using something like a neural architecture search (NAS), which can significantly optimize the search for a well performing architecture for not just the detection step, but for all subsequent steps as well.

7.2.7. Sensing Techniques

This thesis works with the goal of emulating the use of an SLDV as a means of non contact sensing, but there are many more techniques and sensors which could be used for this purpose, like DICs, quantum sensors or even microphones. Investigating how effective each of these sensors is at damage detection and what factors limit them would be interesting research to take on.

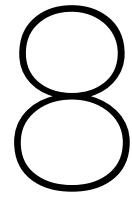
7.2.8. Generalization

Currently, the neural network is trained on data from a coupon sized plate and is tested on the same structure, but it would be interesting to look at whether a network trained on one specific structure can generalize itself to any structure it is used on.

Perhaps it would be interesting to see if, instead of generalizing across different types of structures, the network can generalize at least across one particular type of structure, like plates or beams, could be investigated.

Part IV

Introduction to the GUI



The GUI

8.1. Why a GUI?

Based on the work done throughout the thesis and as explained in the preceding chapters, it was decided that implementing a GUI would be beneficial, as it would allow all steps of the process to be located in one place and be easily accessible to any user who wants to give NN training a go as well as easily visualize the ULS based plots. This chapter will provide a brief overview of the interface, explain some of its capabilities, and show how users can further customise it to suit their needs. The GUI was coded using the PySide6 library, chosen due to its expanded widget selection and layout management compared to the widely used Tkinter library.

Images of all the GUI components have been shown in Appendix A.

8.2. The Main Window

The Main Window is where all the other apps "reside", waiting to be opened. It is a simple window which dynamically resizes itself according to the number of apps in it. Users can freely change the layout and order of the apps, as well as add more apps, as more features are added.

8.3. ULS Viewer

The ULS Viewer app allows users to visualize the datasets by plotting and animating their mode shapes, as well as allowing more such shape based metrics to be visualized. The app has many features, like allowing users to focus on a particular part of the sample instead of the whole thing at a time. It allows users to choose from two formulations for the calculation of the ULS based plots, one the traditional route and the other a weighted approach to give more weight to higher modes.

Based on the formulation, the app calculates Mode Shape, Mode Shape Curvature, ULS Curvature (vs Healthy), and ULS Curvature (vs Healthy) - Mode Shape Curvature, with the option to control the weights and the participating modes. The app also has a symmetry highlight feature to help locate regions where the plot breaks symmetry; additionally users can also export a 3D object of the current plot.

8.4. Node Picker

The Node Picker app is the first part of the damage detection workflow. Here, the user can select which nodes to train the NN on from all scanned nodes in the dataset file. Once the dataset is loaded, the user can set how many nodes they want to pick as an $n \times n$ grid. Currently, it is limited to selecting only $n \times n$ types, but this behaviour can be modified.

Once the selection is made, the user can verify the node coordinates by clicking the *Confirm Selection* button, or select all nodes in the file by clicking the *Select All Nodes* button. Once done, the user must export the selected nodes in the form of a JSON file, which the NN Trainer app will read; an additional file is also stored, which contains the location information of not only the node file but also the dataset, so that the user doesn't have to hassle around trying to find the files repeatedly.

8.4.1. FRF Viewer

On clicking the *See FRFs* button, the user can view the FRF plots for the entire dataset, filtered by sample and selected node. This screen shows four plots, namely the FRFs magnitude, phase, and real & imaginary components with the ability to shift between linear and log plots as well as view only the first n mode to better see the peaks. Alongside this, a toggle exists to create an average healthy FRF from all available healthy samples and overlay it over all the damaged FRFs to easily check for common patterns.

8.4.2. Waterfall Graph Viewer

Apart from the standard FRF view, users can switch to a waterfall graph view which colours peaks and valleys and in essence equate to look at the FRF plot from a top view. The basic plot shows all selected nodes side by side with damaged specimens as well as damaged quadrants being highlighted. Users have many options to tune the plot like which nodes are shown, how many samples are shown at once, how many frequency bins are shown for each graph as well as switching between magnitude, phase, real part and imaginary part.

8.5. Neural Network Trainer

The NN Trainer is the next part of the workflow, which will allow the user to tune and train various NN architectures and visualize their performance. The first section of the app allows the user to select the dataset, node file and output directory of their choice or directly load one of the configuration files saved by the Node Picker.

The 2nd section of the app is the training parameter tuning section, where the user can select input features, training epochs, early stop criteria, batch size, etc. The available training parameters can also be tweaked in code to allow for more granular control.

The 3rd section is the Architecture/ Hyperparameter tuning section. Here, once the user has defined an NN architecture in the prescribed format, they can tune individual hyperparameters without editing the code manually. Once all tuning is complete, the training can be started and tracked using the performance curves, which update as the model trains. At the end of the training, the summary table gets populated with performance metrics of all the folds as well as the overall average and standard deviation. Finally, once the training is completed, the user may choose to save all the plots to a desired location or directly launch the Results Plotter app from within the CNN Trainer or go back to the Main Window and launch it from there.

Bibliography

- [1] "A review of methods and applications in structural health monitoring (SHM) for bridges". *in Measurement*: 245 (2025), **page** 116575. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2024.116575>.
- [2] Osama Abdeljaber **and others**. "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks". *in Journal of Sound and Vibration*: 388 (2017), **pages** 154–170. ISSN: 0022-460X. DOI: <https://doi.org/10.1016/j.jsv.2016.10.043>.
- [3] R. D. Adams **and others**. "A vibration technique for non-destructively assessing the integrity of structures". *in Journal of Mechanical Engineering Science*: 20 (1978), **pages** 93–100.
- [4] Onur Avci **and others**. "A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications". *in Mechanical Systems and Signal Processing*: 147 (2021), **page** 107077. ISSN: 0888-3270. DOI: <https://doi.org/10.1016/j.ymssp.2020.107077>.
- [5] Joshua Ayeni. "Convolutional Neural Network (CNN): The architecture and applications". *in Applied Journal of Physical Science*: 4 (december 2022), **pages** 42–50. DOI: 10.31248/AJPS2022.085.
- [6] Mohsen Azimi, Armin Dadras **and** Gokhan Pekcan. "Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review". *in Sensors*: 20 (may 2020). DOI: 10.3390/s20102778.
- [7] Fray Becerra-Suarez **and others**. "Mathematical Generalization of Kolmogorov-Arnold Networks (KAN) and Their Variants". *in Mathematics*: 13 (september 2025), **page** 30. DOI: 10.3390/math13193128.
- [8] S. Benavides. "Corrosion in the aerospace industry". **in december** 2009: **pages** 1–14. ISBN: 9781845693459. DOI: 10.1533/9781845695538.1.
- [9] Timea Bezdan **and** Nebojsa Bacanin. "Convolutional Neural Network Layers and Architectures". **in january** 2019: **pages** 445–451. DOI: 10.15308/Sinteza-2019-445-451.
- [10] L. Breiman. "Random Forests". *in Machine Learning*: 45 (2001), **pages** 5–32. DOI: 10.1023/A:1010933404324.
- [11] Y. Lu C. Hou. "Identification of cracks in thick beams with a cracked beam element model". *in Journal of Sound and Vibration*: 385 (2016), **pages** 104–124. ISSN: 0022-460X. DOI: <https://doi.org/10.1016/j.jsv.2016.09.009>.
- [12] Gail A Carpenter. "Neural network models for pattern recognition and associative memory". *in Neural Networks*: 2.4 (1989), **pages** 243–257. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(89\)90035-X](https://doi.org/10.1016/0893-6080(89)90035-X).
- [13] P. Cawley **and** R. D. Adams. "The Locations of Defects in Structures from Measurements of Natural Frequencies". *in Journal of Strain Analysis*: 14.2 (1979), **pages** 49–57.
- [14] Maria Nilde Cerri **and** Fabrizio Vestroni. "Use of Frequency Change for Damage Identification in Reinforced Concrete Beams". *in Journal of Vibration and Control*: 9.3-4 (2003), **pages** 475–491. DOI: 10.1177/107754603030787.
- [15] Daily Dose of Data Science. *A Beginner-friendly Introduction to Kolmogorov Arnold Networks (KAN)*. <https://www.dailydoseofds.com/a-beginner-friendly-introduction-to-kolmogorov-arnold-networks-kan/>. Accessed: 17 September 2026. **may** 2024.

- [16] Saranika Das and Koushik Roy. "A State-of-the-Art Review on FRF-Based Structural Damage Detection: Development in Last Two Decades and Way Forward". in *International Journal of Structural Stability and Dynamics*: 22 (december 2021). DOI: 10.1142/S0219455422300014.
- [17] S. W. Doebling and others. *Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in their Vibration Characteristics: A Literature Review*. techreport LA-13070-MS. Los Alamos National Laboratory, 1996.
- [18] Scott W Doebling, Charles R Farrar, Michael B Prime and others. "A summary review of vibration-based damage identification methods". in *Shock and vibration digest*: 30.2 (1998), pages 91–105.
- [19] Finale Doshi-Velez and Been Kim. "Towards a Rigorous Science of Interpretable Machine Learning". in *arXiv preprint arXiv:1702.08608*: (2017). arXiv: 1702.08608.
- [20] Bradley Efron and others. "Least Angle Regression". in *The Annals of Statistics*: 32.2 (2004), pages 407–499. DOI: 10.1214/009053604000000067.
- [21] Olufisayo S. Ekundayo and Absalom E. Ezugwu. "Deep learning: Historical overview from inception to actualization, models, applications and future trends". in *Applied Soft Computing*: 181 (2025), page 113378. ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2025.113378>.
- [22] Theodoros Evgeniou and Massimiliano Pontil. "Support Vector Machines: Theory and Applications". in volume 2049: september 2001, pages 249–257. ISBN: 978-3-540-42490-1. DOI: 10.1007/3-540-44673-7_12.
- [23] David John Ewins. *Modal testing: theory, practice and application*. volume 2. Research studies press Baldock, 2000.
- [24] Nadimul Faisal and others. "Defect Types". in *Structural Health Monitoring Damage Detection Systems for Aerospace*: by editor Markus G. R. Sause and Elena Jasiūnienė. Cham: Springer International Publishing, 2021, pages 15–72.
- [25] C. R. Farrar, N. Dervilis and K. Worden. "The Past, Present and Future of Structural Health Monitoring: An Overview of Three Ages". in *Strain*: 61.1 (2025), e12495. DOI: <https://doi.org/10.1111/str.12495>.
- [26] Charles R Farrar and Keith Worden. "An introduction to structural health monitoring". in *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*: 365.1851 (december 2006), pages 303–315.
- [27] Charles Reed Farrar and Keith Worden. *Structural Health Monitoring: A Machine Learning Perspective*. 2012. URL: <https://api.semanticscholar.org/CorpusID:60745434>.
- [28] Amir Gandomi and others. "Development in Mode Shape-Based Structural Fault Identification Technique". in 5: (january 2008).
- [29] Benyamin Ghojogh and Ali Ghodsi. "Backpropagation and Optimization in Deep Neural Networks". in *Elements of Deep Learning*: Springer Nature Switzerland, 2026, pages 69–94. DOI: 10.1007/978-3-032-10738-1_3.
- [30] S. Hassani and U. Dackermann. "A Systematic Review of Advanced Sensor Technologies for Non-Destructive Testing and Structural Health Monitoring". in *Sensors*: 23.4 (2023), page 2204.
- [31] Kaiming He and others. "Deep Residual Learning for Image Recognition". in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: 2016, pages 770–778. DOI: 10.1109/CVPR.2016.90.
- [32] Hien Hothu and Akira Mita. "Damage Detection Method Using Support Vector Machine and First Three Natural Frequencies for Shear Structures". in *Open Journal of Civil Engineering*: 03 (january 2013), pages 104–112. DOI: 10.4236/ojce.2013.32012.

- [33] Tao Huang and Mykola Bobyr. "A Review of Delamination Damage of Composite Materials". in *Journal of Composites Science*: 7 (november 2023), page 468. DOI: 10.3390/jcs7110468.
- [34] Gerhard Hübschen. "Ultrasonic Testing". in *Materials Characterization Using Nondestructive Evaluation (NDE) Methods*: by editor Gerhard Hübschen and others. Woodhead Publishing, 2016, pages 113–145. ISBN: 978-0-08-100411-1. DOI: 10.1016/B978-0-08-100411-1.00005-X.
- [35] Nathan Ida and Norbert Meyendorf. *Handbook of advanced nondestructive evaluation*. volume 10. Springer International Publishing Cham, Switzerland, 2019.
- [36] Daniel Irawan, Evgeny V. Morozov and Murat Tahtali. "Machine Learning–Based Method for Structural Damage Detection". in *Data Science in Engineering Vol. 10*: by editor Thomas Matarazzo and others. Cham: Springer Nature Switzerland, 2025, pages 137–142.
- [37] S. Edwards J.K Sinha M.I Friswell. "SIMPLIFIED MODELS FOR THE LOCATION OF CRACKS IN BEAM STRUCTURES USING MEASURED VIBRATION DATA". in *Journal of Sound and Vibration*: 251.1 (2002), pages 13–38. ISSN: 0022-460X. DOI: <https://doi.org/10.1006/jsvi.2001.3978>.
- [38] Ho-Yeon Jung, Seung-Hoon Sung and Hyung-Jo Jung. "Experimental Validation of Normalized Uniform Load Surface Curvature Method for Damage Localization". in *Sensors*: 15.10 (2015), pages 26315–26330. ISSN: 1424-8220. URL: <https://www.mdpi.com/1424-8220/15/10/26315>.
- [39] George Kardomateas and Philippe Geubelle. "Fatigue and Fracture Mechanics in Aerospace Structures". in december 2010: ISBN: 9780470686652. DOI: 10.1002/9780470686652.eae142.
- [40] N.T. KHIEM and T.V. LIEN. "A SIMPLIFIED METHOD FOR NATURAL FREQUENCY ANALYSIS OF A MULTIPLE CRACKED BEAM". in *Journal of Sound and Vibration*: 245.4 (2001), pages 737–751. ISSN: 0022-460X. DOI: <https://doi.org/10.1006/jsvi.2001.3585>.
- [41] Edyta Kowalska. "Damage Identification of a Steel Plate Using Vibration Methods". in *Civil and Environmental Engineering Reports*: 34 (november 2024), pages 360–371. DOI: 10.59440/ceer/195900.
- [42] Ernst Kussul and others. "Rosenblatt perceptrons for handwritten digit recognition". in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*: volume 2. IEEE. 2001, pages 1516–1520.
- [43] Shiyun Liao and others. "A channel-spatial-temporal attention-based network for vibration-based damage detection". in *Information Sciences*: 606 (2022), pages 213–229. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2022.05.042>.
- [44] Zachary C. Lipton. "The Mythos of Model Interpretability". in *Queue*: 16.3 (2018), pages 31–57. DOI: 10.1145/3236386.3241340.
- [45] Ziming Liu and others. "KAN: Kolmogorov-Arnold Networks". in *arXiv preprint arXiv:2404.19756*: (2024). DOI: 10.48550/arXiv.2404.19756.
- [46] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". in *Advances in neural information processing systems*: 30 (2017).
- [47] J. Luo, M. Huang and Y. Lei. "Temperature Effect on Vibration Properties and Vibration-Based Damage Identification of Bridge Structures: A Literature Review". in *Buildings*: 12.8 (2022), page 1209.
- [48] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". in *The Bulletin of Mathematical Biophysics*: 5.4 (1943), pages 115–133. DOI: 10.1007/BF02478259.

- [49] Sidharth Mishra **and others**. “Principal Component Analysis”. in *International Journal of Livestock Research*: (january 2017), **page** 1. DOI: 10.5455/ijlr.20170415115235.
- [50] T. Q. Nguyen **and others**. “Application of CNN and MLP models for structural health monitoring: A case study on Saigon Bridge”. in *Journal of Low Frequency Noise, Vibration and Active Control*: 44.3 (2025), **pages** 1825–1854. DOI: 10.1177/14613484251332711.
- [51] Zhaoyang Niu, Guoqiang Zhong **and** Hui Yu. “A review on the attention mechanism of deep learning”. in *Neurocomputing*: 452 (2021), **pages** 48–62. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2021.03.091>.
- [52] Kehinde Olu-lawal **and others**. “INNOVATIVE MATERIAL PROCESSING TECHNIQUES IN PRECISION MANUFACTURING: A REVIEW”. in *International Journal of Applied Research in Social Sciences*: 6 (march 2024), **pages** 279–291. DOI: 10.51594/ijarss.v6i3.886.
- [53] M. Pástor, Michal Binda **and** Tomáš Harčarik. “Modal Assurance Criterion”. in *Procedia Engineering*: 48 (december 2012), **pages** 543–548. DOI: 10.1016/j.proeng.2012.09.551.
- [54] B Peeters **and others**. “Evaluation of structural damage by dynamic system identification”. in *Proceedings of the International Seminar on Modal Analysis*: Katholieke Universiteit Leuven. 1996, **pages** 1349–1362.
- [55] “Quantitative identification of damage in composite structures using sparse sensor arrays and multi-domain-feature fusion of guided waves”. in *Measurement*: 208 (2023), **page** 112482. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2023.112482>.
- [56] Fabrizio Ricci **and others**. “Ultrasonic Guided Waves for Hidden Damage Detection in Composite Structures: Theory and Experiments”. in *RILEM Bookseries*: 6 (january 2012), **pages** 577–584. DOI: 10.1007/978-94-007-0723-8_82.
- [57] Cynthia Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. in *Nature Machine Intelligence*: 1.5 (2019), **pages** 206–215. DOI: 10.1038/s42256-019-0048-x.
- [58] Cynthia Rudin **and others**. “Interpretable machine learning: Fundamental principles and 10 grand challenges”. in *Statistics Surveys*: 16 (2022), **pages** 1–85.
- [59] A. Rytter, M. Krawczuk **and** P. H. Kirkegaard. “Experimental and Numerical Study of Damaged Cantilever”. in *Journal of Engineering Mechanics*: 126.1 (2000), **pages** 60–65. DOI: 10.1061/(ASCE)0733-9399(2000)126:1(60).
- [60] Anders Rytter. *Vibrational Based Inspection of Civil Engineering Structures*. English. Ph.D.-Thesis defended publicly at the University of Aalborg, April 20, 1993 PDF for print: 206 pp. Denmark, 1993.
- [61] O.S. Salawu. “Detection of structural damage through changes in frequency: a review”. in *Engineering Structures*: 19.9 (1997), **pages** 718–723. ISSN: 0141-0296. DOI: [https://doi.org/10.1016/S0141-0296\(96\)00149-6](https://doi.org/10.1016/S0141-0296(96)00149-6).
- [62] Muhammad Shafiq **and** Zhaoquan Gu. “Deep Residual Learning for Image Recognition: A Survey”. in *Applied Sciences*: 12.18 (2022).
- [63] Jonathon Shlens. *A Tutorial on Principal Component Analysis*. 2014. arXiv: 1404.1100 [cs.LG]. URL: <https://arxiv.org/abs/1404.1100>.
- [64] Hoon Sohn. “Effects of environmental and operational variability on structural health monitoring”. in *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*: 365.1851 (2007), **pages** 539–560. DOI: 10.1098/rsta.2006.1935.
- [65] S. Somvanshi **and others**. “A Survey on Kolmogorov-Arnold Network”. in *arXiv preprint arXiv:2411.06078*: (2024). arXiv: 2411.06078.

- [66] William Soo Lon Wah, Yung-Tsang Chen **and** John S Owen. "A regression-based damage detection method for structures subjected to changing environmental and operational conditions". *in***Engineering Structures**: 228 (2021), **page** 111462. ISSN: 0141-0296. DOI: <https://doi.org/10.1016/j.engstruct.2020.111462>.
- [67] Iirina Trendafilova **and** Ward Heylen. "Fault localization in structures from remote FRF measurements. Influence of the measurement points". *in*(**january** 1998).
- [68] Turn2Engineering Editorial Team. *Failure Mechanisms: How Materials Fail*. <https://turn2engineering.com/mechanical-engineering/materials-science/failure-mechanisms>. Accessed: 16 September 2026. **may** 2026.
- [69] Eric Udd, **editor**. *Fiber Optic Smart Structures and Skins II*. **volume** 1170. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. **february** 1990.
- [70] H. Van der Auweraer **and** B. Peeters. "International research projects on structural health monitoring: An overview". *in***Structural Health Monitoring**: 2.4 (2003), **pages** 341–358.
- [71] Z. Wang, R.M. Lin **and** M.K. Lim. "Structural damage detection using measured FRF data". *in***Computer Methods in Applied Mechanics and Engineering**: 147.1 (1997), **pages** 187–197. ISSN: 0045-7825. DOI: [https://doi.org/10.1016/S0045-7825\(97\)00013-3](https://doi.org/10.1016/S0045-7825(97)00013-3).
- [72] Keith Worden **and** Graeme Manson. "The application of machine learning to structural health monitoring". *in***Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**: 365.1851 (**december** 2006), **pages** 515–537.
- [73] Law S.S Wu. D. "Damage localization in plate structures from uniform load surface curvature". *in***Journal of Sound and Vibration**: 276.1 (2004), **pages** 227–244. ISSN: 0022-460X. DOI: <https://doi.org/10.1016/j.jsv.2003.07.040>.
- [74] C. B. Yun **and others**. "Recent Research and Application Activities on Structural Health Monitoring in Korea". *in***Journal of Civil Engineering Research and Practice**: 1.1 (2004), **pages** 1–12.
- [75] Chunyan Zeng **and others**. "Squeeze-and-Excitation Self-Attention Mechanism Enhanced Digital Audio Source Recognition Based on Transfer Learning". *in***Circuits, Systems, and Signal Processing**: 44 (**september** 2024), **pages** 480–512. DOI: 10.1007/s00034-024-02850-8.

Part V

Appendix

A

GUI Images

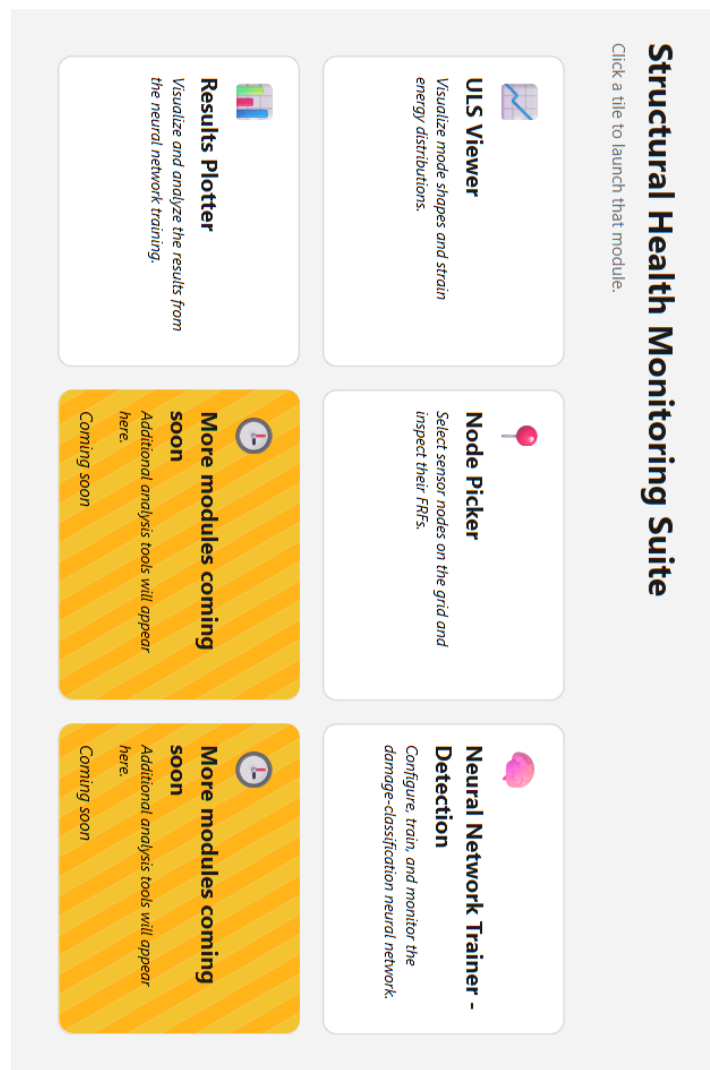
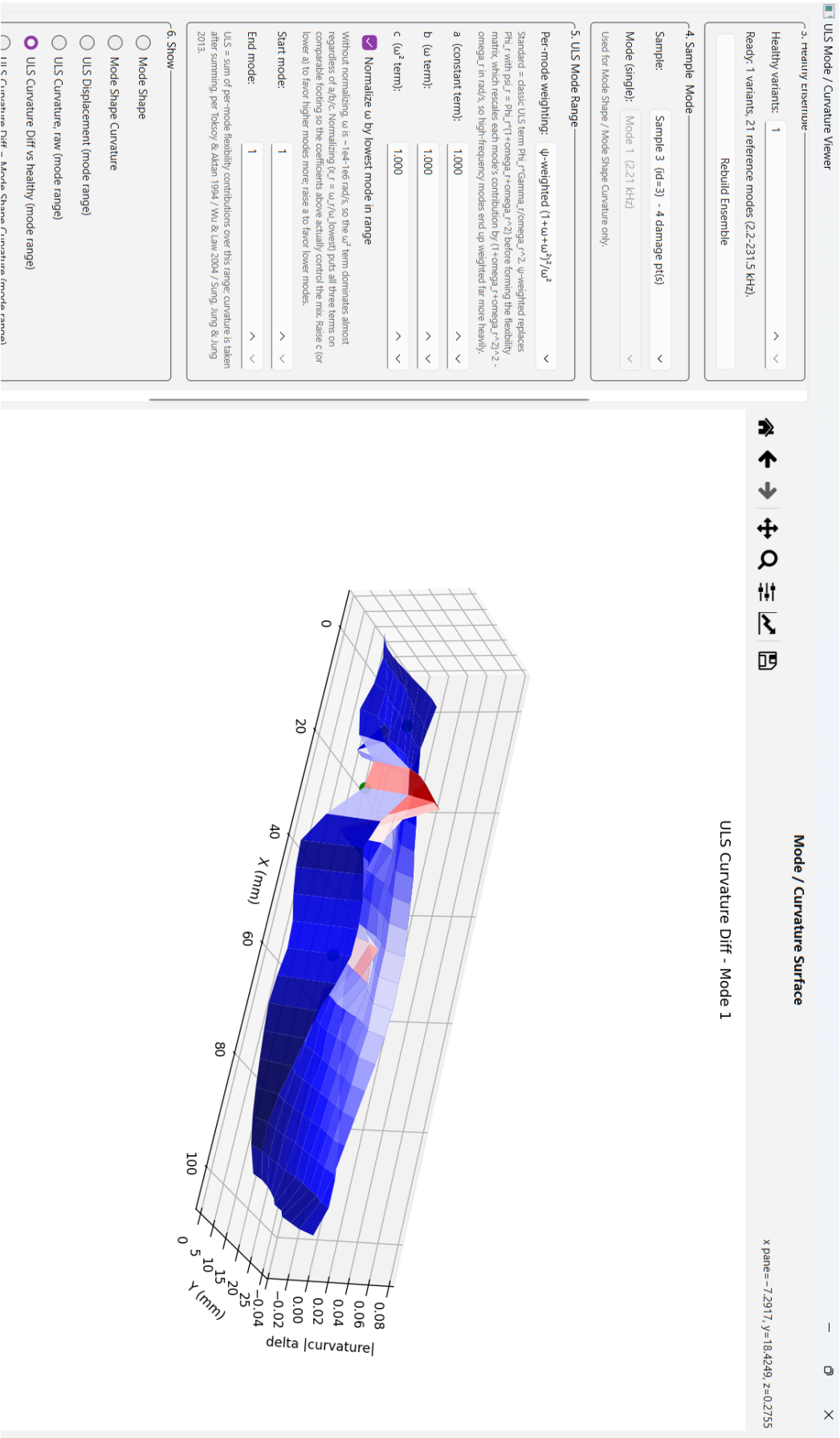


Figure A.1: Main window of the GUI



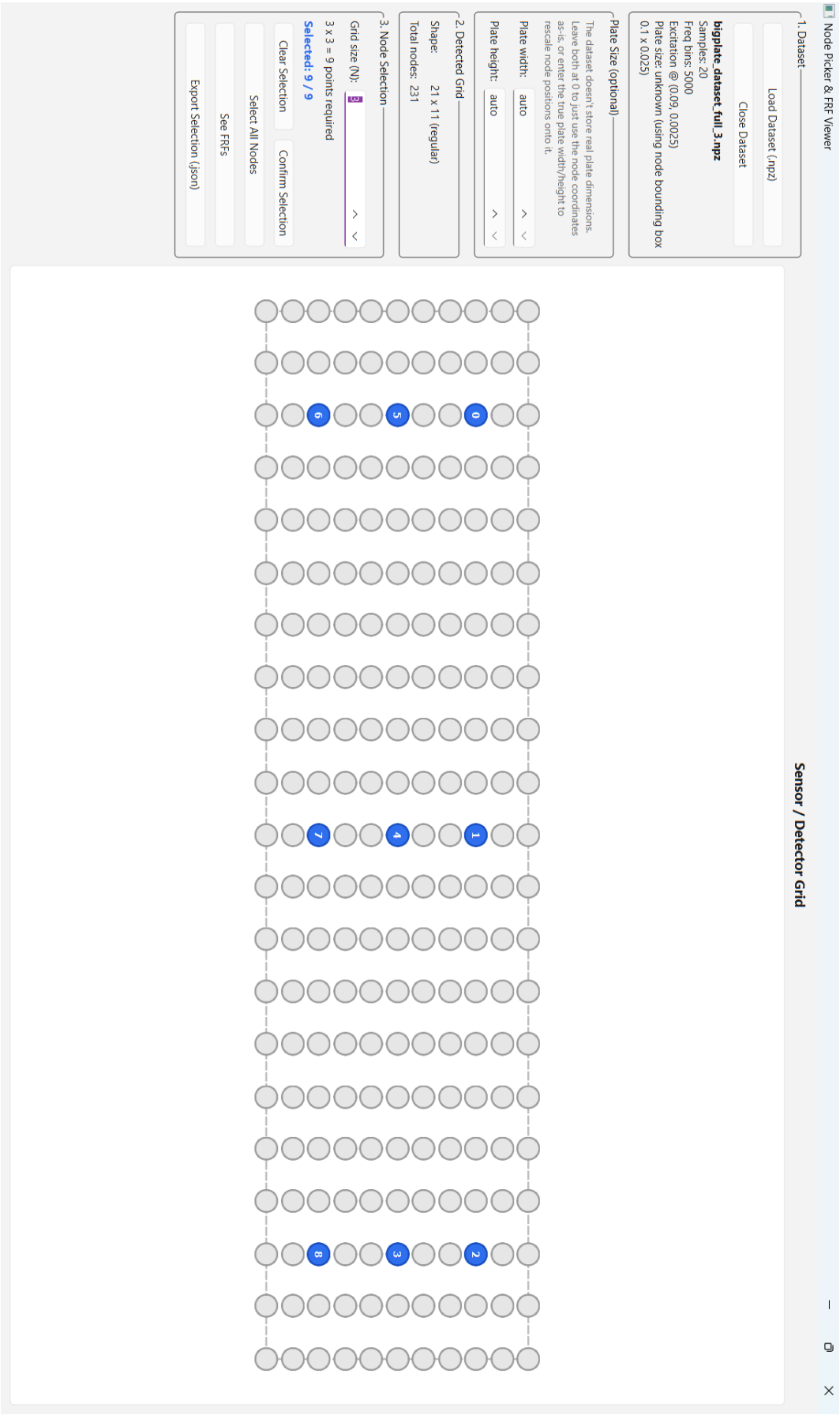


Figure A.3: Main screen of the Node Picker app

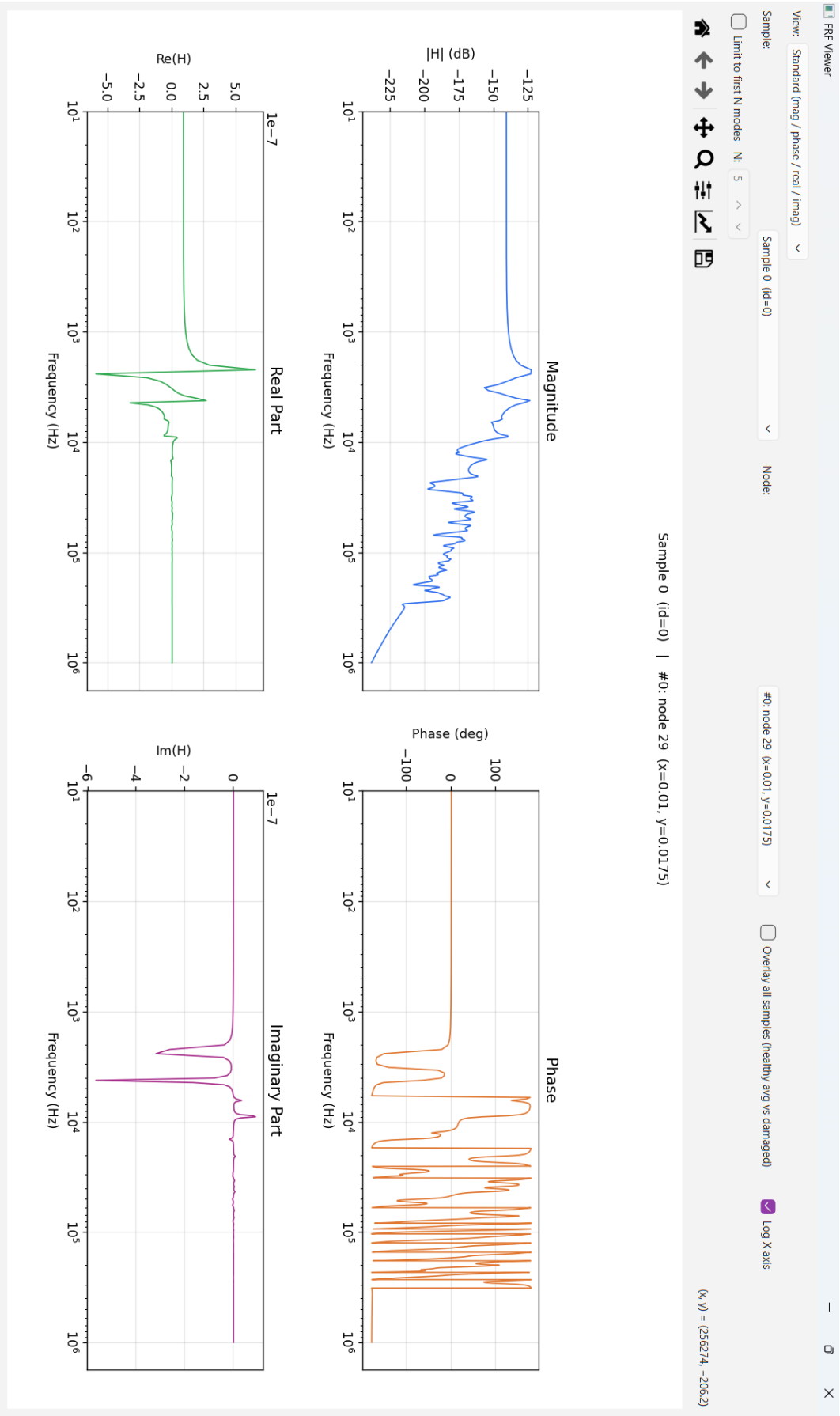


Figure A.4: FRF viewing screen of the Node Picker app



Figure A.5: Waterfall graph viewing screen

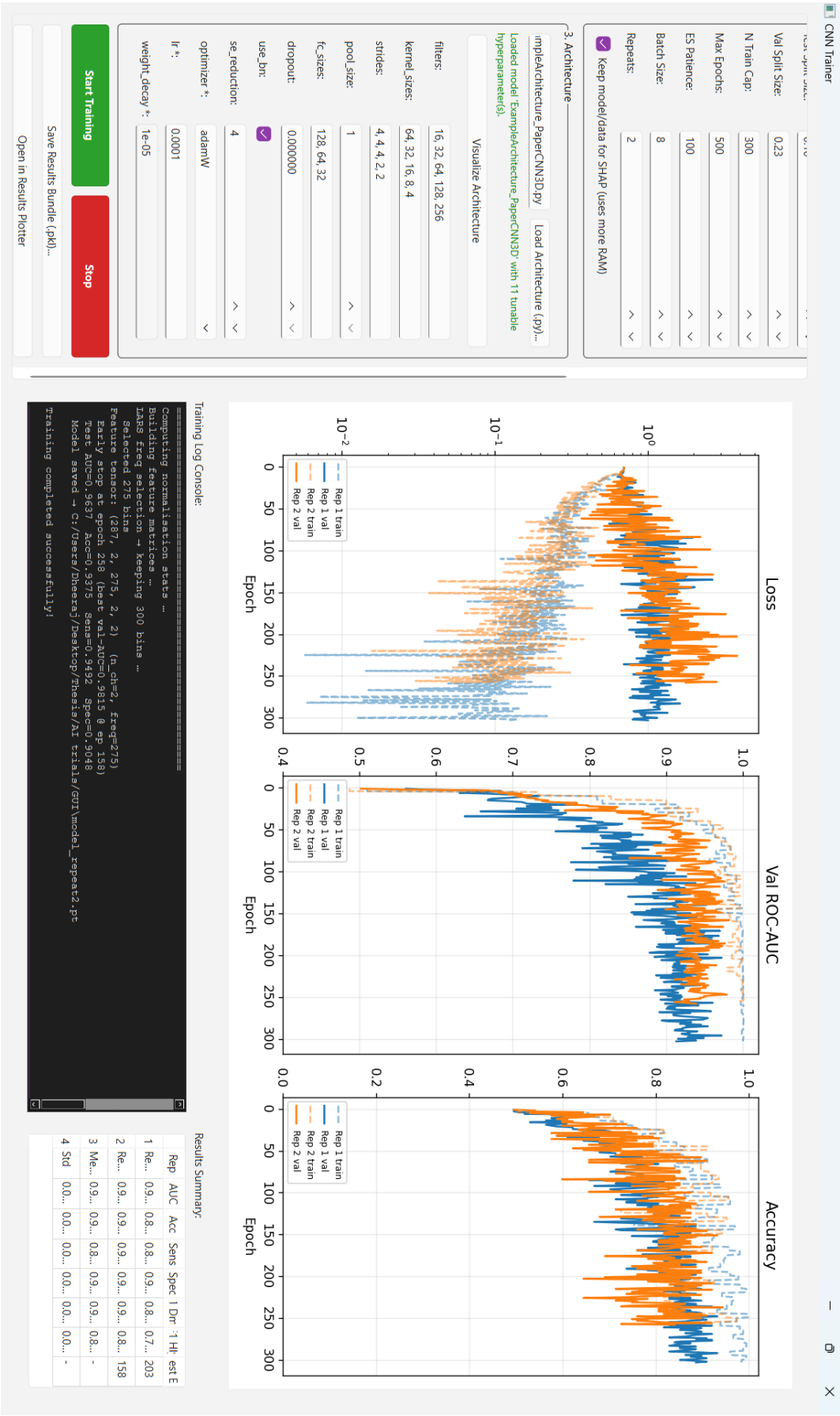


Figure A.6: CNN Trainer app

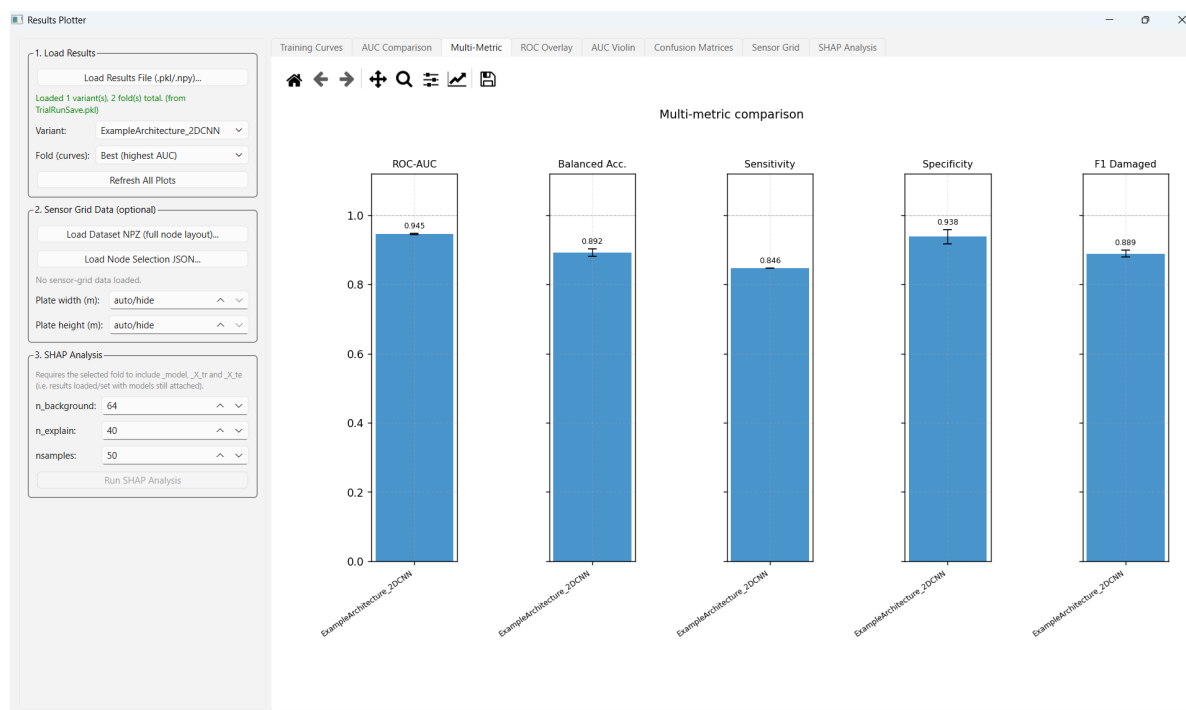


Figure A.7: Result Plotter app showing Multi-Metric plot

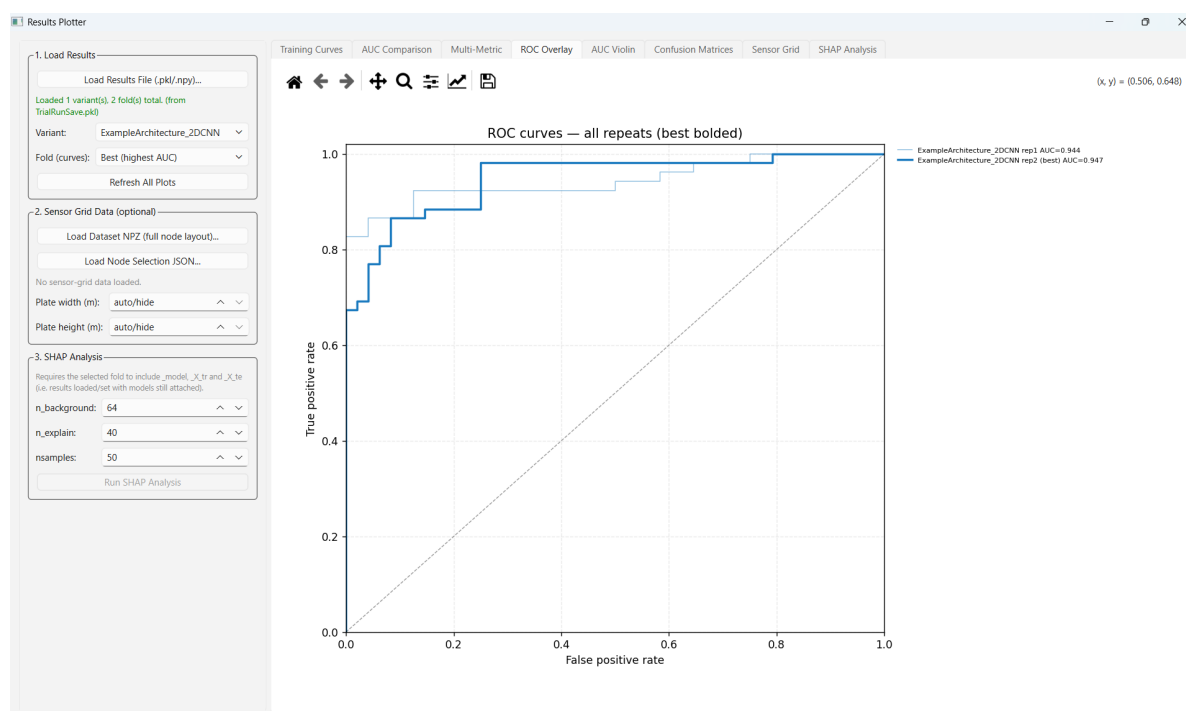


Figure A.8: Result Plotter app showing ROC curve plot

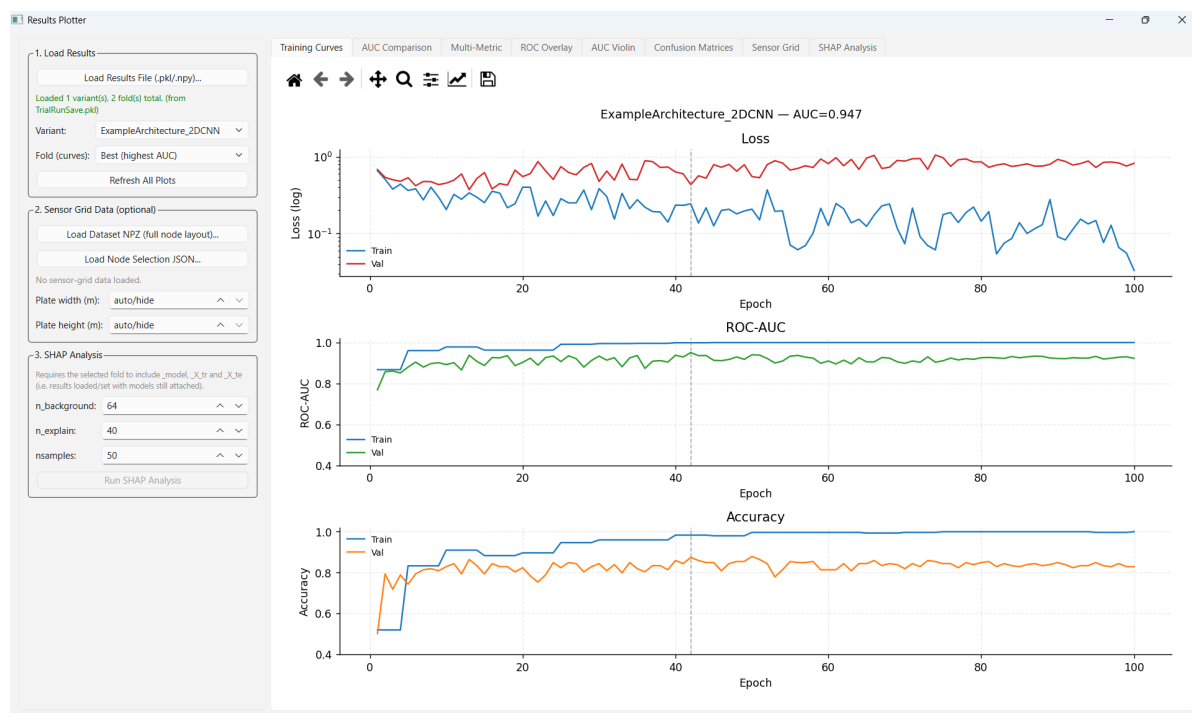


Figure A.9: Result Plotter app showing Learning curves plot

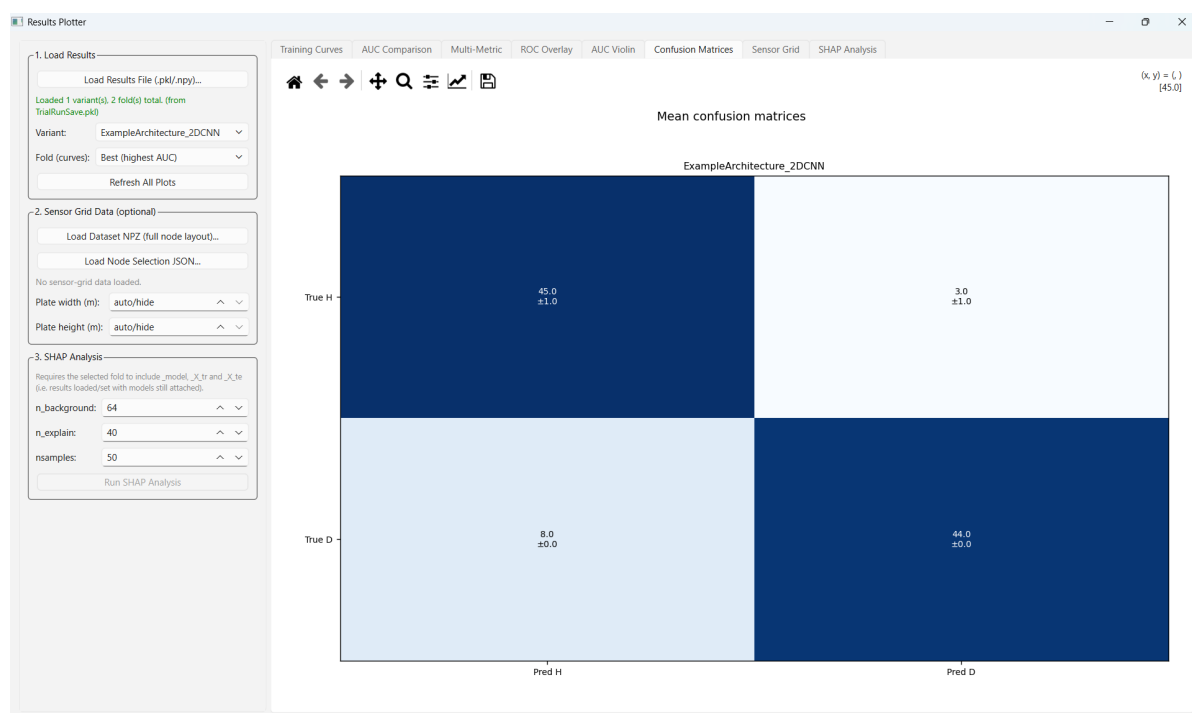


Figure A.10: Result Plotter app showing Confusion matrix plot

