machine learning & knowledge extraction

MDPI

Article

Transforming Simulated Data into Experimental Data Using Deep Learning for Vibration-Based Structural Health Monitoring

Abhijeet Kumar [1,*], Anirban Guha [1] and Sauvik Banerjee [2]

[1] Department of Mechanical Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India; anirbanguha@iitb.ac.in
[2] Department of Civil Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India; sauvik@civil.iitb.ac.in
* Correspondence: abhijeet.k@iitb.ac.in

Abstract: While machine learning (ML) has been quite successful in the field of structural health monitoring (SHM), its practical implementation has been limited. This is because ML model training requires data containing a variety of distinct instances of damage captured from a real structure and the experimental generation of such data is challenging. One way to tackle this issue is by generating training data through numerical simulations. However, simulated data cannot capture the bias and variance of experimental uncertainty. To overcome this problem, this work proposes a deep-learning-based domain transformation method for transforming simulated data to the experimental domain. Use of this technique has been demonstrated for debonding location and size predictions of stiffened panels using a vibration-based method. The results are satisfactory for both debonding location and size prediction. This domain transformation method can be used in any field in which experimental data for training machine-learning models is scarce.

Keywords: domain transfer; structural health monitoring; vibration; deep leaning; experimental data generation

check for updates

Citation: Kumar, A.; Guha, A.; Banerjee, S. Transforming Simulated Data into Experimental Data Using Deep Learning for Vibration-Based Structural Health Monitoring. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 18–40. https://doi.org/10.3390/make6010002

Academic Editor: Andreas Holzinger

Received: 6 November 2023
Revised: 13 December 2023
Accepted: 21 December 2023
Published: 27 December 2023

1. Introduction

Data-driven methodologies in the field of structural health monitoring (SHM) have seen significant success in a wide range of applications, including damage detection, localization and quantification. The data-based SHM system uses machine learning (ML), pattern-recognition techniques, or statistical techniques to identify structural conditions based on observed data [1,2]. By extracting meaningful patterns and knowledge from large datasets, data mining enables improved damage-related detection, diagnosis, prognosis and decision-making in SHM systems. Over the last three decades, numerous data-driven approaches for SHM have been reported for various structures employed within the civil, aerospace and mechanical domains using vibration, ultrasonic, electrical and magnetic-signal based approaches [3–7]. Out of these, vibration and ultrasonic signals have been very commonly used for SHM because of their effectiveness and wider applicability for damage detection in structures [8,9]. There are generally two types of machine learning (ML) algorithms: supervised learning and unsupervised learning. A supervised-learning approach involves training the learning algorithm with available data and a label. The structure's undamaged or damaged state, the location of the damage, and the severity of the damage are all possible labels in SHM. For the supervised learning approach, all damage severity and locations of interest for a specific structure must be included in the training dataset. The unavailability of experimentally recorded data from a real structure with all relevant class labels is the main problem with supervised learning. In the absence of experimentally recorded damage data for any structure, one can also use unsupervised machine learning. However, this model's limitation is that it can only identify the presence or absence of damage [10]. The model does not offer any specific damage information, such as damage location or severity. One needs to incorporate supervised learning in SHM in order to perform damage diagnosis at an advanced level.

In the majority of cases, collecting training data for the data-based approach through experiments is not practical due to time and resource constraints. Alternatively, the generation of training data using a physics-based model and testing with experimental data could be an alternative strategy. However, this approach could be problematic because the underlying distributions of the data generated from the physics-based model and the data from the actual structure are different for various reasons, such as environmental uncertainty in the experiment, modelling bias of the physics-based model, etc. ML models, whether supervised or unsupervised, are created with the presumption that testing and training data demonstrate the same distribution. Generation of experimental data through a large number of experiments for the purpose of training an ML model is considered impractical. Due to these issues, data-based approaches are not widely used in SHM. These issues are particularly problematic for supervised learning, because it needs labeled damage-state data for all potential damage scenarios. Nevertheless, supervised learning has been used for the past three decades to diagnose damage in SHM [11,12]. The development of a model for the diagnosis of damage using a data-driven method that trains with data generated by a physics-based model and tests with actual signals captured from a structure by means of experiments has gained attention in recent years. A strain field-monitoring-based diagnostic system has been shown to detect, localize and quantify anomalies in a real crack propagating on an aluminum stiffened-skin structure [13]. Lamb wave signals and a numerically-enhanced ML approach have been used for assessment of cracks in plate-like structures [14]. Lamb-wave-scattering-based hole-type defect diagnosis has been reported in laminated composites [15,16]. Application of domain adaptation has also been reported in SHM [17]. A combination of simulated and experimental data achieved through deep-metric-learning-based SHM has reportedly been used for guided wave signals [18]. Application of artificial neural network and Probability Ellipse techniques for Lamb-wave-based damage detection has also been reported [19].

From the study of past work, it is evident that several efforts have been made to mitigate the unavailability of real-structure-based damage data for data-driven SHM. However, most of the previous studies have considered guided wave data. There is a clear lack of similar studies related to vibrational-data-based SHM. In this work, we attempt to fill this lacuna. We report a more generalized methodology than described in a previously reported work [20], in order to reduce the variability between numerically simulated data and experimental data, so that one can generate training data by numerical simulation. In the proposed method, numerically simulated data has been transferred to the experimental domain, as shown in Figure 1.

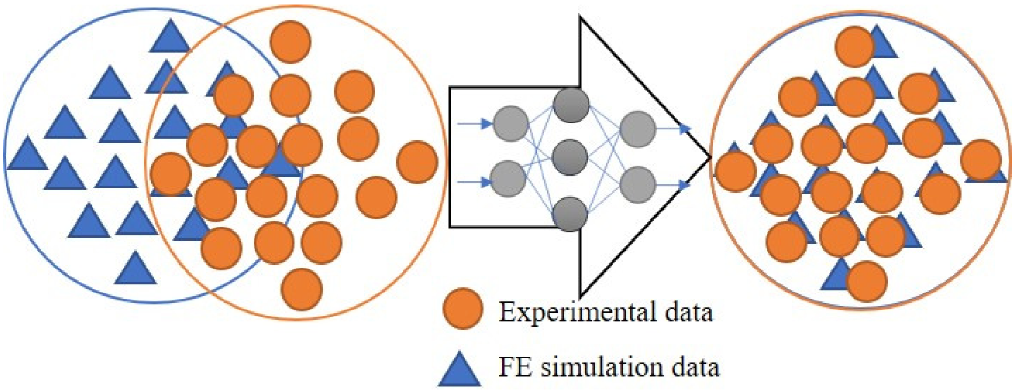


Figure 1. Schematic representation of domain transfer.

Once the simulated database is transformed to the experimental domain, the transformed dataset is used to train the machine-learning model for damage assessment. Thereafter, the trained ML model is used to estimate the damage condition using experimental data. For the domain transformation, we have used an artificial neural network (ANN)-based deep learning (DL) model, and for damage assessment we used a stacked learning model. The proposed methodology is employed to determine debonding localization and quantification in metallic stiffened panels, based on vibration analysis.

The paper is organized into the following sections: in Section 2, the proposed methodology is described; in Section 3, the experimental set-up and specimen details are provided; in Section 4 the Finite Element (FE) simulation of stiffened panels is described; in Section 5, the detailed implementation of the proposed methodology is discussed, and Section 6 gives concluding remarks.

2. Methodology

Related Works

In data-driven SHM, addressing the lack of data recorded from actual structures is a crucial challenge. Researchers have investigated numerous methods and strategies for addressing this issue. Table 1 lists some important contributions to the tackling of the lack of availability of damage data recorded from real structures. Some of the work reported in the literature considering the vibration-based approach either uses a relatively high amount of experimental data or studies very simple structures. For example, Barthorpe et al. created various damage scenarios by adhering and detaching panels for the wing panel of an airplane wing taken from a single experimental specimen [21]. Similarly, Bao et al. created various damage scenarios by loosening and tightening a bolt on a single portal frame [22]. However, this is not possible for many other forms of structures. Sbaruffatti et al. introduce the scale factor to reduce experimental and FE model biases [13]. The FE model is then used to generate various damage-scenario datasets for the ML model's training, and these datasets are tested with experimental data. The author has conducted extensive study on the evaluation of fatigue damage in helicopter fuselage structures. The presented method produces satisfactory results for damage assessment. Due to the difference in boundary condition fixity between the FE models and the experiments the model bias is also very prevalent in vibration models. Consequently, the present work is motivated by scaling the FE simulated data to experimental data as demonstrated by Sbaruffatti et al. [13]. In this study, rather than directly scaling FE data to experimental data, we use a deep learning model to transfer the FE domain to an experimental domain for vibration-based SHM.

Table 1. Related work addressing the lack of available damage data in SHM.

Authors	**Method**	**Approach**	**Structure**	**Damage Type**	**Exp. Data Quantity**
Sbarufatti et al., 2013 [13]	Scaling factor	Strain field	Helicopter fuselage	Fatigue damage	Less exp. data
Sbarufatti et al., 2013 [14]	Numerically-enhanced ML approach	Guided wave	Plate	Discontinuity	Less exp. data
Barthorpe and Worden 2017 [21]	Generating exp. data from one specimen	Vibration	Wing panel of airplane	Discontinuity	High exp. Data
Gardner and Worden 2019 [17]	Domain adaptation	Vibration	MDOF spring-mass body	Stiffness variation	Less exp. Data
Zhang et al., 2022 [18]	Effective combination of FE and exp. data	Guided wave	Plate	Discontinuity	Equal quantity of FE and exp. data
Bao et al., 2023 [22]	Transfer Learning	Vibration	Portal frame	Nut–bolt loosening	Relatively more exp. Data

This section provides a summary of the entire domain-transfer-based damage assessment methodology (Figure 2). The goal is to review all of the fundamental components of the procedure and demonstrate how they were put together to achieve the current investigation's goal, which is the implementation of an SHM system based on model-enhanced signal processing. The proposed methodology consists of three major parts.

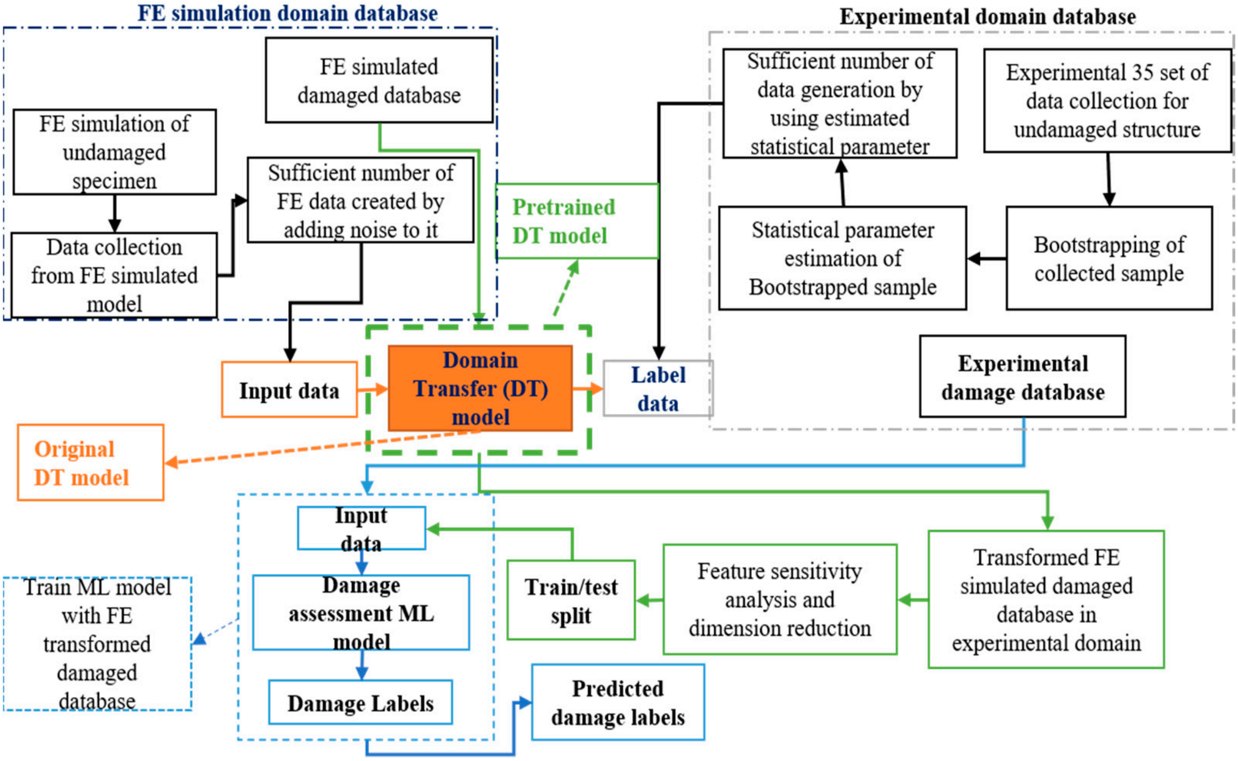

FE simulation domain database
FE simulated damaged database
FE simulation of undamaged specimen
Sufficient number of FE data created by adding noise to it
Data collection from FE simulated model
Pretrained DT model
Experimental domain database
Sufficient number of data generation by using estimated statistical parameter
Experimental 35 set of data collection for undamaged structure
Statistical parameter estimation of Bootstrapped sample
Bootstrapping of collected sample
Input data
Domain Transfer (DT) model
Label data
Experimental damage database
Original DT model
Input data
Damage assessment ML model
Damage Labels
Train ML model with FE transformed damaged database
Train/test split
Feature sensitivity analysis and dimension reduction
Transformed FE simulated damaged database in experimental domain
Predicted damage labels

Figure 2. Domain-transfer-based SHM methodology framework.

Part one is experimental data pre-processing, part two is domain-transformation model architecture definition and part three is damage assessment using transformed dataset. The objective of the first part is to obtain the statistical distribution of the relevant parameters from the experimental data. The goal of the second part is to construct a neural network model which effectively transforms the FE-simulated data into the experimental domain. The third part deals with the feature-sensitive analysis, dimension reduction and proper ML model selection for damage assessment.

The first part of the proposed methodology begins with collecting a minimum of 30 (to obtain sufficient statistical parameter estimation according to the central limit theorem) samples of experimental data of an undamaged structure. In the present work, we have collected 35 samples of experimental data. Then the collected data were resampled by bootstrapping [23]. Bootstrapping is, broadly defined, expressed by the law of large numbers; this implies that if original samples are resampled repeatedly with replacement, then the resampled data would closely reflect the actual population data. Bootstrapping techniques have been used in many scientific studies for resampling or for better statistical accuracy with the sample data [24–27]. Figure 3 shows the bootstrap sampling procedure used in the present study. The bootstrap sample was then examined with quantile–quantile (Q-Q) plots to see whether it had the same distribution as the original data.

In the second part, the objective is to construct a model which accurately transforms the FE-simulated database into the experimental domain. The schematic diagram of the transformation model is shown in Figure 4. The input dataset for the model is undamaged FE-simulated model data and the output is the experimental dataset generated in part one of the methodology.

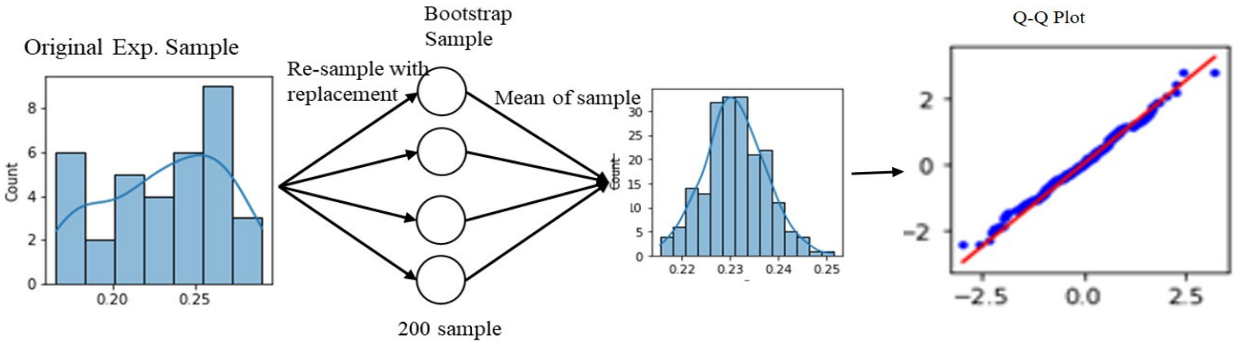


Figure 3. Bootstrap sampling procedure and sampling distribution estimation using a Q-Q plot.

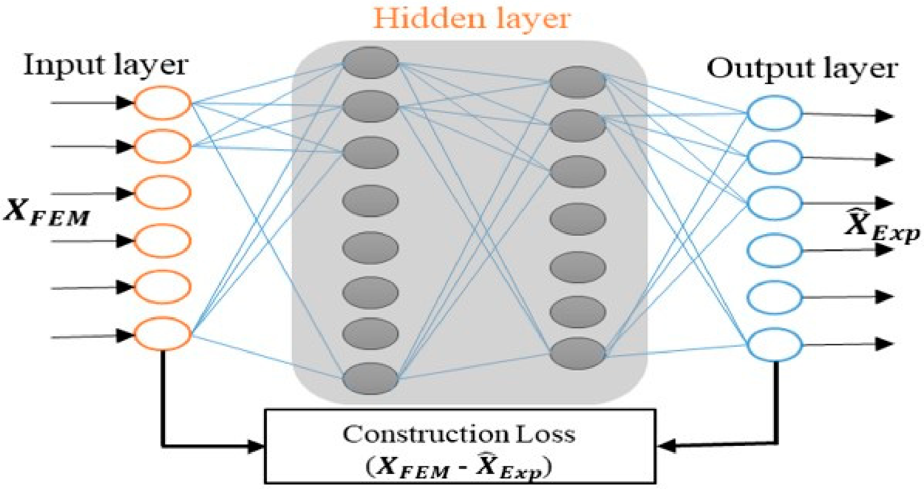


Figure 4. Schematic domain transfer model.

The third part of methodology (Figure 5) begins with a reduction in the dimensionality of the transformed feature. In this work, we use a two-step dimension reduction process. In step one, the sensitivity of the feature to be studied for damage is measured using the Mahalanobis square distance (MSD). The features are filtered using a threshold MSD value. Then, the dimensionality for the filtered features is further reduced using principal component analysis (PCA). This is followed by training and performance evaluation through ML algorithms. In the final step, the trained ML algorithm is tested with an experimental dataset.

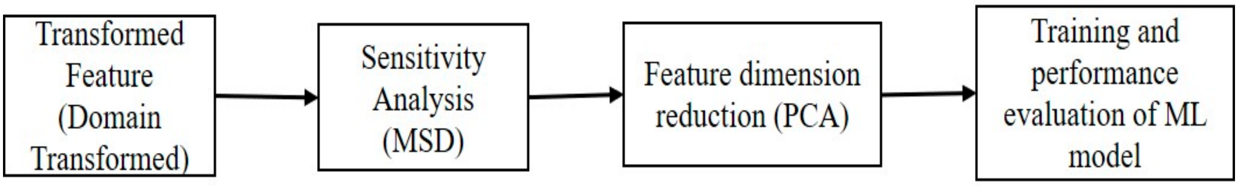

Transformed Feature (Domain Transformed)
Sensitivity Analysis (MSD)
Feature dimension reduction (PCA)
Training and performance evaluation of ML model

Figure 5. Sequence of order of transformed dataset to damage assessment.

3. Experimental Setup and Specimen Detail

In the present work, a scanning laser Doppler vibrometer (LDV) was used to extract vibration data for metallic stiffened panels. The experimental setup for the LDV (manufactured by PolyTec, Germany) consists of a laser scanning head (PVS 500), an LDS magnetic oscillator (V406 M4-CE) manufactured by LDS, England, a function generator (TEKTRONIX 3021B) manufactured by Tektronix, USA and a voltage amplifier (PA100E CE) manufactured by LDS, England, as shown in Figure 6.

The experiment was completed in the following sequential steps: positioning of scan head, 2-D alignment, definition of scan points on the specimen, laser focus, scan, and data extraction.

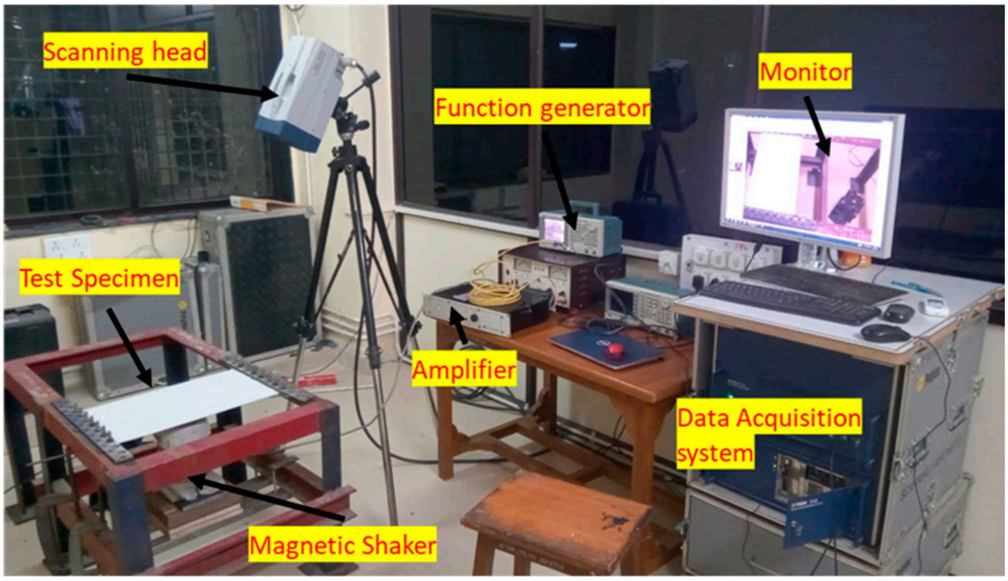


Figure 6. Experimental set-up of the LDV.

3.1. Test Specimen

The experiment was carried out on a metallic stiffened panel made of aluminum plate and a T-stiffener of material grade AL5052 H32. The dimensions of the plate are 530 mm length, 330 mm width and 1.5 mm thickness, and the stiffener has web dimensions of 26 mm × 2 mm and flange dimensions of 2 mm × 26 mm. Epoxy adhesive was used to affix the stiffener to the plate. The epoxy adhesive was made by combining resin and hardener in equal amounts (Loctite EA E-102). The bonding–curing time was 24 h. Four specimens were created, one undamaged and the other three damaged (with debonding). Table 2 shows the locations of the damaged points.

Table 2. Test Specimens.

Specimen 1	Specimen 2	Specimen 3	Specimen 4
Intact (Undamaged)	Debonding at X = 265 of L = 50	Debonding at X = 265 of L = 100	Debonding at X = 285 of L = 70

Where X is the center of the debonding location, and L is the length of the debonding in mm.

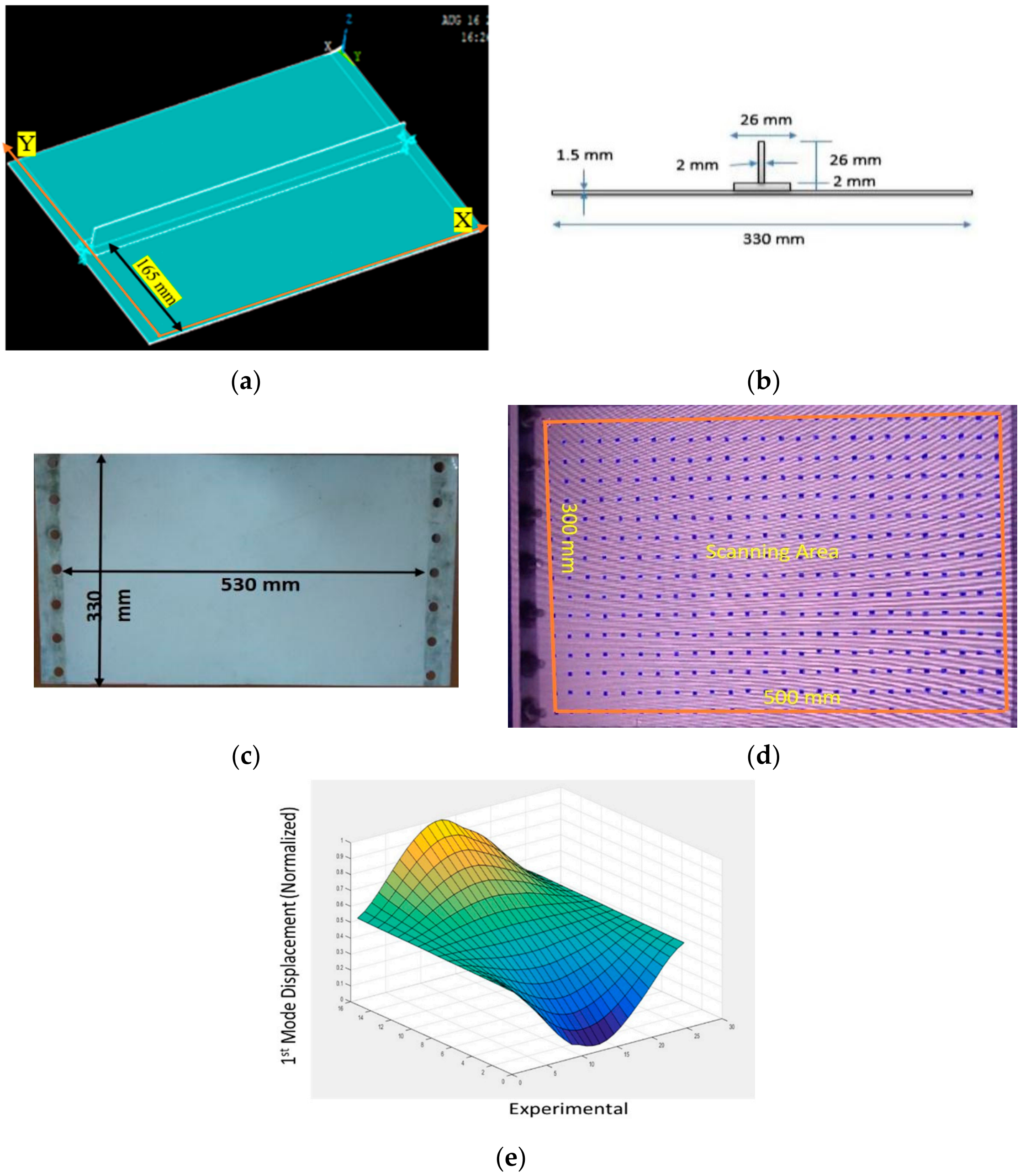
To improve the laser beam's reflection from the stiffened structure, the plate is painted white (RUST-OLEUM 249126 FLAT WHITE). There were a total of 416 node points, 26 along the length and 16 along the width, with a 20 mm distance between each (Figure 7). To prevent laser beam scattering, only 300 mm × 500 mm out of 530 mm × 330 mm is taken into account when measuring a point's area, as shown in Figure 7d. Beyond this area, the incident laser beam might extend past the specimen due to the bending of the specimen during the experiment.

To measure the modal frequency of the experimental specimen, the parameters of the LDV were as given in Table 3.

Table 3. Experimental parameters.

Parameters	Value
Minimum speed	5 μm/s
Maximum speed	10 m/s
Frequency resolution	0.125 Hz
Frequency range	0–200 Hz
Number of FFT lines	12,800
Scan time for one scan point	64 s

FFT: fast Fourier transform.



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

Figure 7. (**a**) Specimen top view; (**b**) cross-section of stiffened panel; (**c**) white-painted backside of specimen; (**d**) defined scan points; (**e**) 1st mode shape.

3.2. Experimental Variability Reduction

One of the most difficult aspects of conducting an experiment is ensuring that the results are consistent with those from a repeated testing of the same specimen or while testing a different specimen. The major source of experimental variability in vibration-based experiments is related to fixed boundary conditions. Maintaining the same boundary condition for all specimens is nearly impossible. However, maintaining equal clamping forces can help minimize the variability of boundary conditions. To accomplish this, at the fixed boundary condition, all clamping nut-and-bolt arrangements were tightened with a torque wrench to a constant torque. The sequence in which the bolts were changed was also considered, with a consistent attachment sequence maintained throughout the test. The mean of 11 tests was used as a single value of experimental data for each specimen. This was implemented while bootstrapping the original sample's data.

4. Numerical Modeling of Stiffened Panel

The finite-element software package ANSYS 16.1 was used to model a stiffened aluminum plate with the same dimensions as the experimental specimen. For the modelling of the plate and stiffener, the eight-node quadrilateral solid shell element (SOLSH190) with three degrees of freedom in translation direction at each node was used. To determine the size of the element, a convergence analysis was performed and, finally, meshing of the entire structure was performed with 1 mm element size. The mechanical properties of the modelled specimen are E = 70.3 GPa, density = 2680 kg/m^3 and Poisson ratio = 0.33. Debonding is simulated by breaking the node connection between the plate and stiffener. The contact in the area of debonding is modelled using the parameters in Table 4.

Table 4. Contact modelling parameters.

Parameter	Type
Contact type	Flexible surface-to-surface
Target element	TARGE170
Contact element	CONTA174
Contact algorithm	Augmented Lagrange method
Location of contact detection	Gauss point
Contact selection	Asymmetric
Gap/closure	No adjustment
Behavior of the contact surface	No separation
Geometry	3-D

Numerical Model Verification with Experiment

A numerical model's ability to recreate a real test sample is limited. Nonetheless, certain critical panel parameters can be verified. The important parameters of the vibration-based model are modal frequency and mode shape displacements. However, it is very difficult to verify the mode shape displacements in this specific case of a stiffened plate. An attempt was therefore made to verify the undamaged specimen's modal frequency. Changing the clamping torque value modifies the fixed boundary condition. This can be utilized to obtain a modal frequency of an experimental test sample that is close to the numerical model. At a torque of 22 Nm, the experimentally determined first modal frequency showed good agreement with the frequency value predicted by the finite element model, with a difference of 1.17%. Table 5 shows the first three modal frequencies that were obtained from FE-simulated models and experiments on both undamaged and damaged samples.

Table 5. Model frequency (Hz).

	Undamaged			**50 mm Debonding**			**100 mm Debonding**		
Mode	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
FEM	49.95	70.00	110.26	49.91	69.73	109.43	49.91	69.34	109.43
Exp.	49.37	68.22	105.9	49.296	68.12	105.51	49.12	67.86	104.95

5. Methodology Implementation

5.1. Data Preparation for Domain Transformation

The experimental data pre-processing framework discussed in the methodology section is used to create the experimental dataset for domain transfer. The data preparation started with the performance of 35 experiments on the LDV experimental set-up to collect 35 samples (as per the central limit theorem, to have a strong statistical estimation, we require at least 30) of data for each scanning point (shown in Figure 7d) of the undamaged stiffened panel. Here, it should be noted that we have a total of 416 scanning points, as shown in Figure 7d. Because of the symmetry of the stiffened panel along its length, we only used half of the points, i.e., the first 208 scanning points, in this study, as shown in Figure 8. The 35 data points were bootstrapped to create a larger sample size of 200. To reduce the experimental variability of the experimental data, one element of bootstrap data is the mean of 11 elements of the originally collected data, as discussed in Section 3.2. Figure 9a shows the histogram of the initial 35 data points, Figure 9b shows the 200 bootstrapped sample data points, and Figure 9c shows the quantile–quantile (Q-Q) plot for the bootstrap sample data. The Q-Q plot confirms that the bootstrapped data has the same distribution as that of the original sample. Subsequently, the mean and standard deviation (SD) of this sample were calculated and were used with the Python-based NumPy library, which was employed to generate a sample of size 1000 for every scan point.

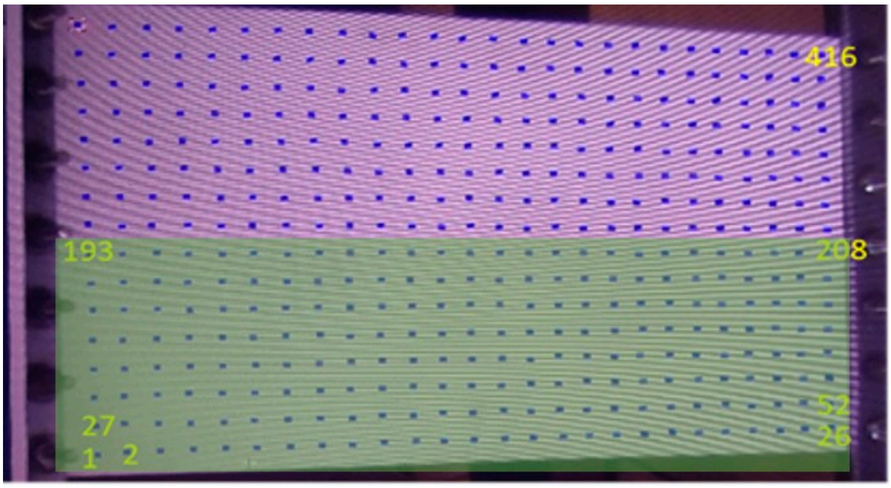
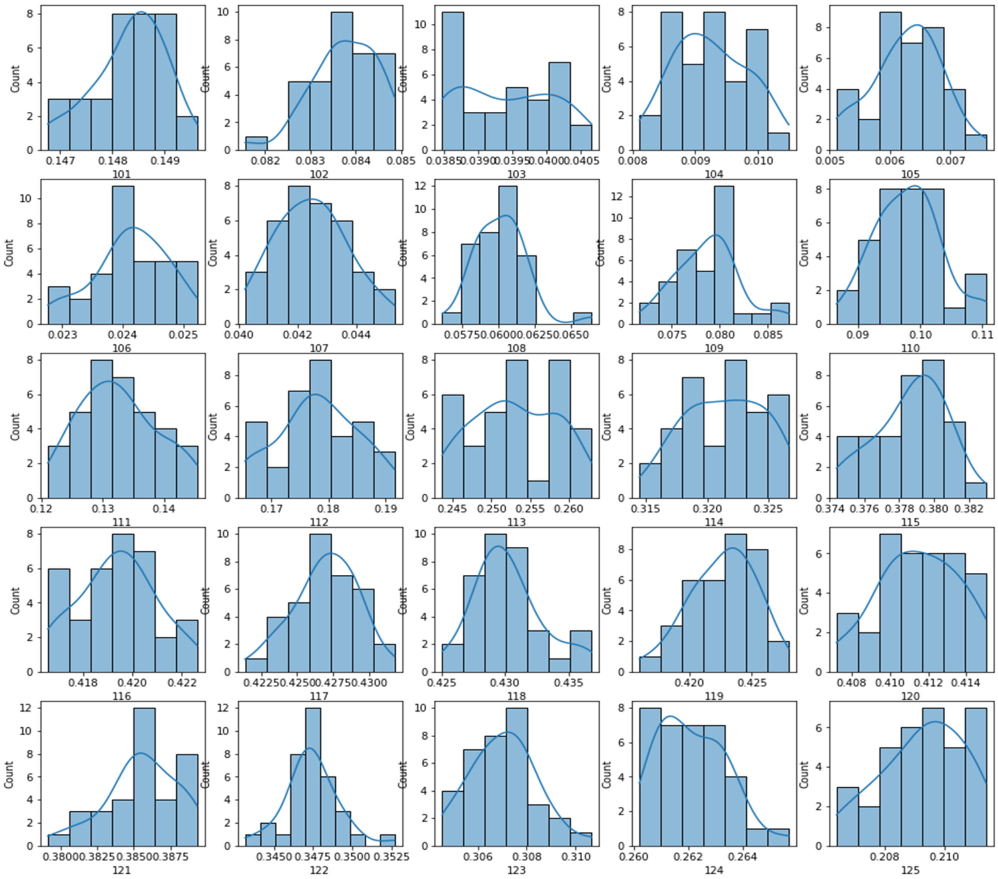
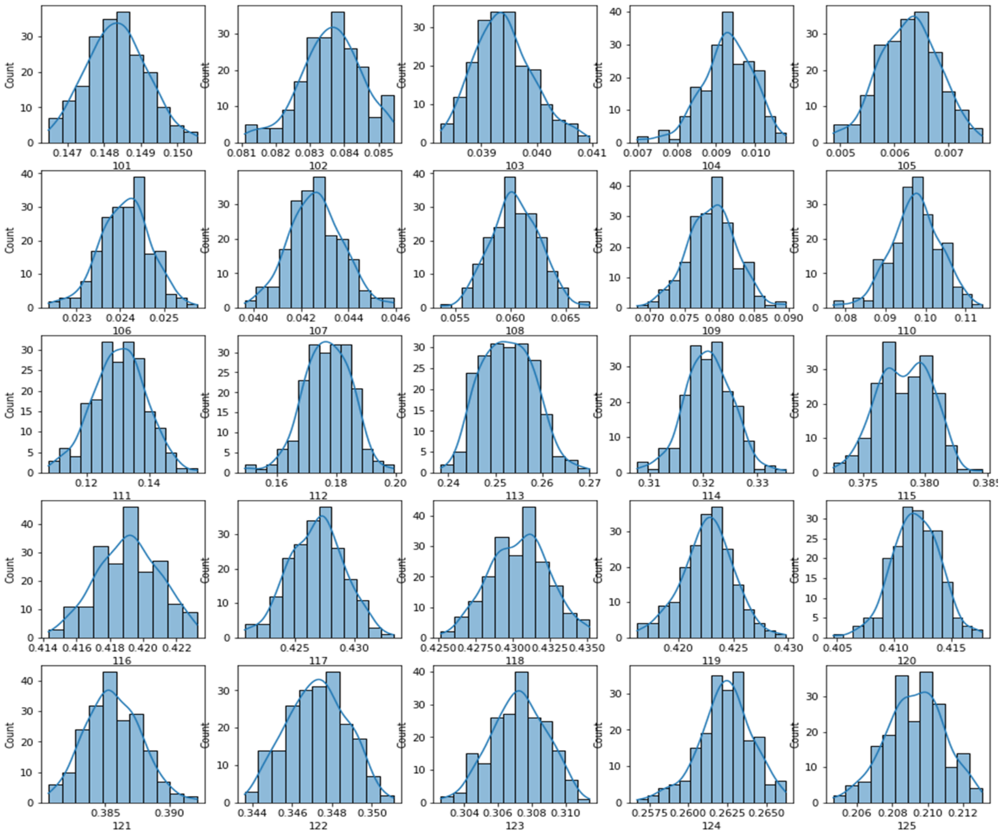


Figure 8. Symmetry of the stiffened panel.

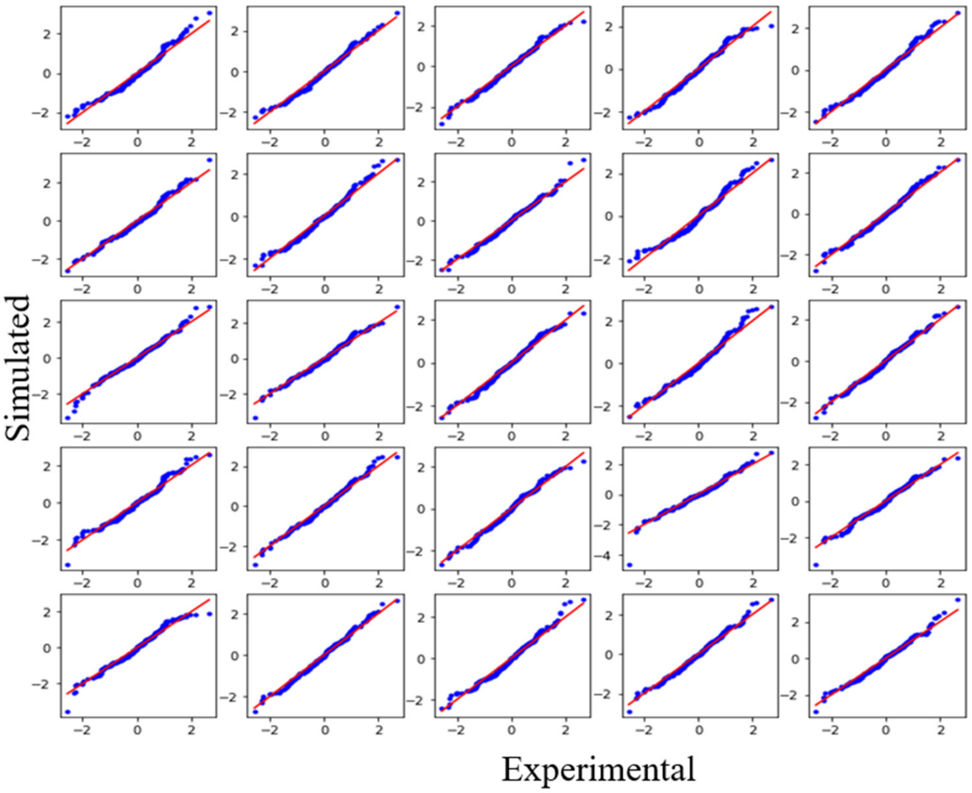


(**a**)

Figure 9. *Cont.*



(**b**)



(**c**)

Figure 9. (**a**) Initial 35 points of collected data; (**b**) 200 points of bootstrapped sample data; (**c**) Q-Q plot for normal distribution.

Similarly, we also need 1000 sets of undamaged FE-simulated data. In order to create 1000 pieces of data, FE-simulated data were repeated 1000 times, adding 0.2 percent random Gaussian noise to each instance through the Python based NumPy library. Now we have two 1000 × 208 matrices, one for the undamaged sample experiment, and one for the FE simulation of the same.

5.2. Domain Transformation Models

For choosing the appropriate domain transformation model, three neural network-based models were constructed, namely, an artificial neural network (ANN), a one-dimensional convolution neural network (1D-CNN) and a 1D-CNN with a fully connected network (FC) (1D-CNN-FC-CNN), as models 1, 2 and 3, respectively. ANNs are computer programs that work like the neural network in the brain. An ANN is made up of a group of nodes or neurons. Neurons are connected to each other, making a network structure similar to that of biological neural mechanisms, which is where intelligence comes from. An ANN has a set number of neurons in each layer: input, hidden and output. Mathematically, the ANN for input X and label $\hat{X}$ can be represented as

$$\hat{y} = \sigma\left(W^T X + b\right) \tag{1}$$

where $\hat{y}$ is predicted output, X is input vector, W is weight, b is bias and σ is an activation function.

A loss function compares the target output value and the predicted output value to determine how effectively the ANN models the training data. During training, the goal is to reduce this difference between the predicted and desired outputs as much as possible. The mean square error (MSE) loss function is defined as

$$L(\hat{y}, \hat{X}) = \|\hat{y}^{(i)} - \hat{X}^{(i)}\|^2 \tag{2}$$

Here, L is loss.

The objective of an ANN training algorithm is to minimize the loss function by optimizing the value of weight W and bias b, defined as

$$J(W^T, b) = \frac{1}{M}\sum_{i=1}^{M} L(\hat{y}^{(i)}, \hat{X}^{(i)}) \tag{3}$$

Here, J is the Jacobian which optimizes the weight W and bias b to minimize loss; L, M is the number of samples in the training dataset.

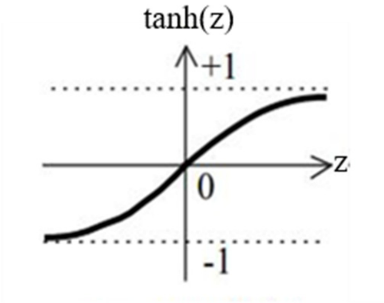
The 1D-CNN is an effective deep learning tool used for finding the underlying pattern in a one-dimensional sequence dataset. Several damage detection methods [28–30] have used it in different fields. A standard CNN has the following parts: the input layer, convolution layers, a pooling layer and a fully connected layer. Each convolution layer is made up of kernels $k(m)$ of length m whose parameters can be learned. Mathematically, the convolution layer for input $x(n)$ of length n and kernel $k(m)$ is shown as [31]

$$y(n) = \rho\left(x(n) \bigotimes k(m)\right) \tag{4}$$

$$\rho(n) \bigotimes k(m) = \sum_{k=0}^{n} x(k)k(n-m) \tag{5}$$

Here, ρ represents the activation function, while the symbol $\otimes$ denotes the convolution operation, m is kernel length and n is input length. To obtain an outcome from convolution, a pooling layer is utilized which extracts inherent signal features.

The standard activation functions in ANN architecture are the tangent hyperbolic (tanh), rectified linear unit (ReLU), leaky ReLU, and sigmoid. The tangent hyperbolic (Figure 10) showed better performance in terms of sensitivity and learning speed than the other activation functions in the initial trials. Therefore, we have chosen tanh as the activation function to be used throughout the present study. To further accelerate the training process, batch normalization (BN) is introduced between the CNN layers. The training of deep neural nets is significantly accelerated by BN as it tries to minimize the internal covariate shift. It achieves this by fixing the means and variances of the layer inputs during the normalization step [32]. The model's architecture and its parameters are shown in Figure 11. The detailed parameters of the model are provided in Appendix A. The total number of trainable parameters for the three models are 845520, 2118112 and 2905641 for models 1, 2 and 3, respectively.



$$\sigma = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Figure 10. Tanh activation function.

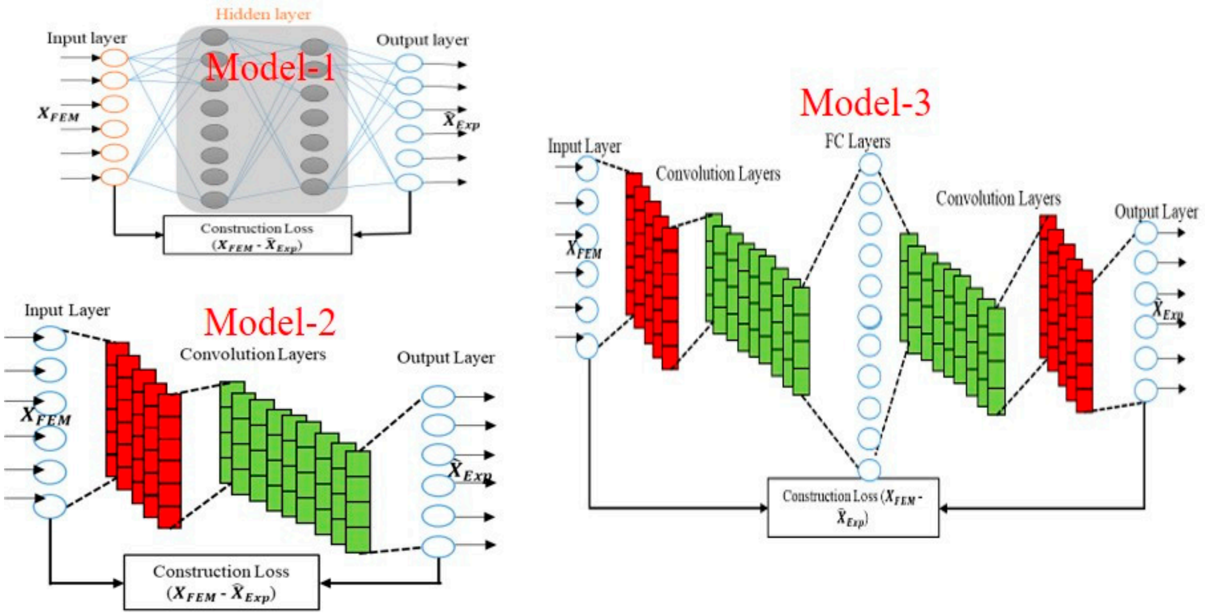
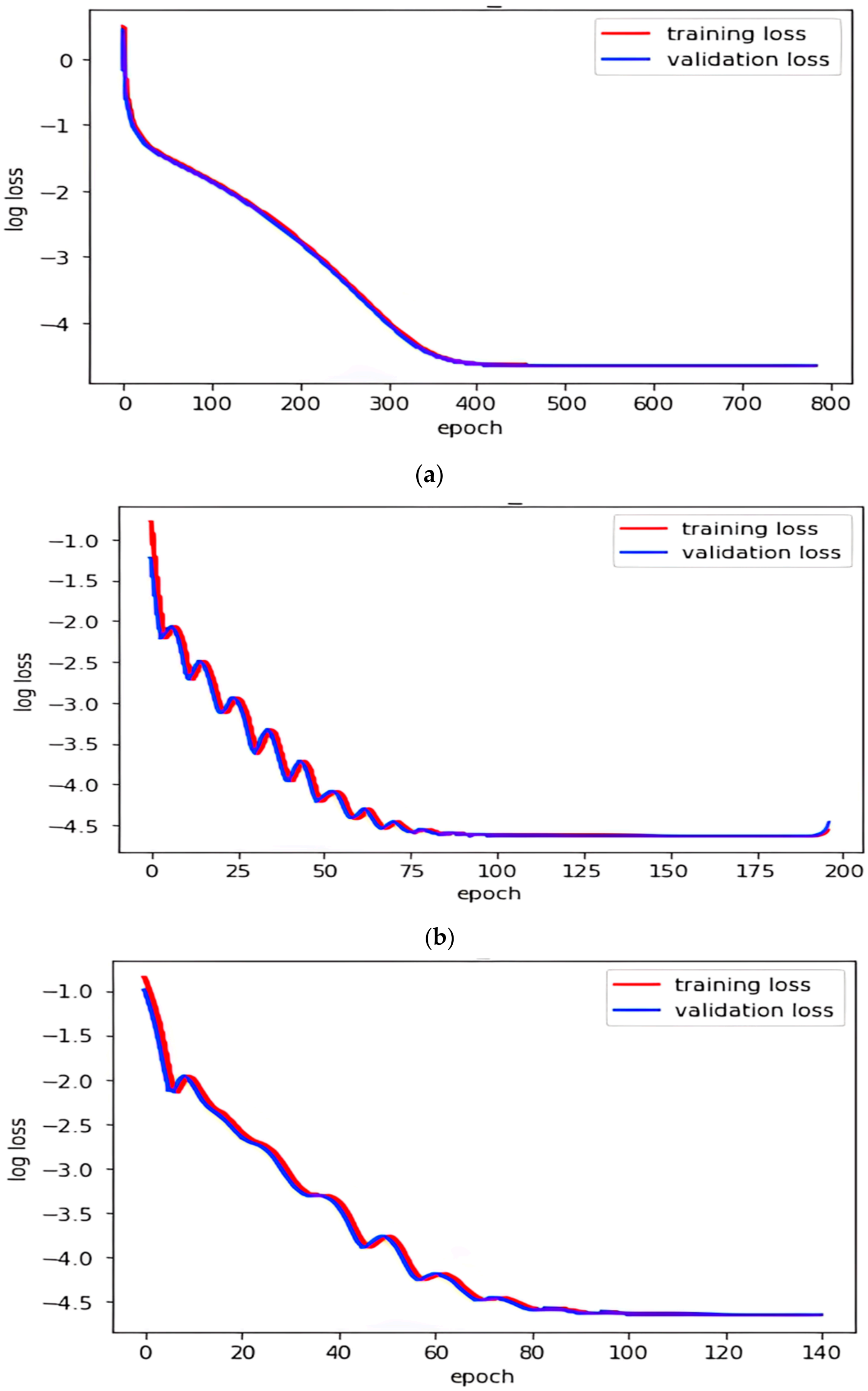


Figure 11. Types of domain transformation model: Model-1 as a deep neural network, Model-2 as a 1D-CNN, and Model-3 as a 1D-CNN-FC-CNN.

Once the domain transformation (DT) models are constructed, the next step is the training of the models. The source domain D_s (input) and target domain D_t (output) for the DT model are FE-simulated database X and an experimental database $\hat{X}$, both with a size of 1000 × 208. The DT model was trained with the following parameters: batch size of 200, learning rate of 0.0002, MSE loss function and adaptive momentum (Adam) optimizer with a weight decay of 1×10^{-5}. In order to prevent the problem of overfitting, an early stopping criterion that guarantees generalization has been adopted [33]. Pytorch (version 1.11), a deep learning library that is based on Python, is used in both the construction and training of the DT models. The entire process of the methodology pipeline was implemented using a local computer (Dell OptiPlex 7070) with the following configuration: device type, CPU (4.6 GHz); processor, core 6 intel i7 8700; memory-16 GB; and OS, Windows 10. Figure 12 shows the training progression of the DT models.



(**a**)

(**b**)

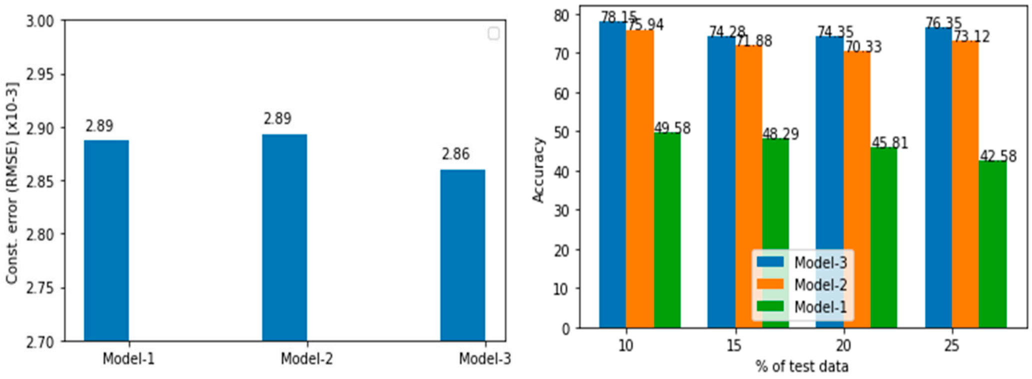
(**c**)

Figure 12. Training processes of the DT models in terms of epoch vs. log loss for (**a**) model-1; (**b**) model-2; (**c**) model-3.

5.3. Comparison of DT Models

After the DT model construction and training, the performance of the models was compared. There are two relevant criteria that were set: the first was how accurately the model transformed the FE-simulated data to the experimental domain, and the second was to check the effectiveness of the transformed dataset. Here, effectiveness is defined as the ability of the transformed dataset to give better prediction accuracy in damage assessment.

The first criterion is evaluated by comparing the average root mean square error (RMSE) between the damaged-specimen experimental data and the damaged-specimen FE simulation data after transformation for each damaged experimental specimen. The model with the lowest RMSE is considered the best. Here, it is important to mention the experimental damage data preparation procedure. We collected 15 sets of experimental data for each damaged specimen by performing 15 experiments on the LDV lab setup. Then, a procedure was followed similar to that used for the undamaged-structure experimental data pre-processing, as discussed in Section 5.1. Finally, 50 data points were generated for each damaged-specimen case. Figure 13a shows the average RMSE of the DT models while transforming FE-simulated damaged-specimen data to the experimental domain for each point of experimental damaged-specimen data. The figure clearly shows that DT model 3 produces the least RMSE in transforming the FE data to the experimental domain.



(**a**) (**b**)

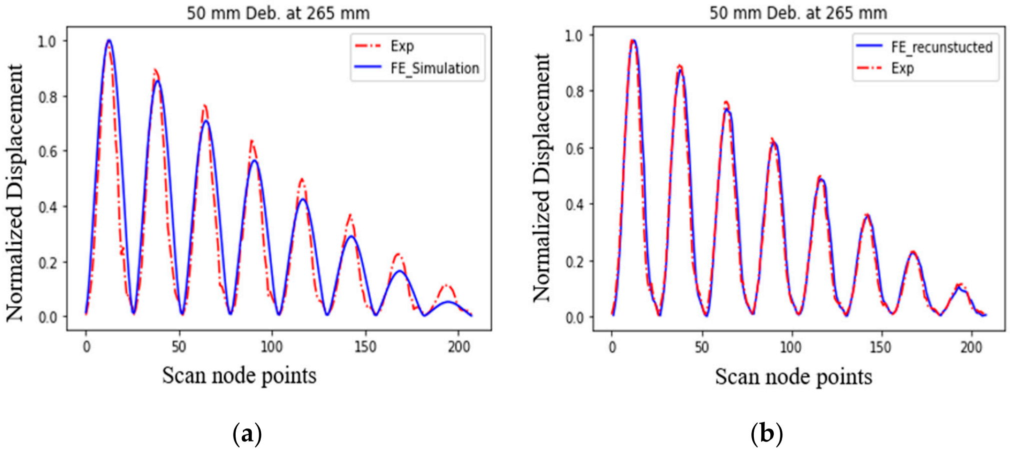
Figure 13. (**a**) Average construction RMSE; (**b**) ML model debonding localization performance on FE-transformed dataset using DT models 1, 2 and 3.

To check the second criteria, i.e., the effectiveness of the transformed dataset, first, the FE-simulated damaged-specimen database was transformed into an experimental domain using the DT models. Then, feature dimensions of the transformed datasets were reduced using PCA. The number of PCA components was determined by checking the cumulative sum of variance and then using the number of components which capture 98% variance in the dataset. Then, this PCA of the transformed dataset was split into training and testing datasets, followed by the training of an ML model with the training dataset and testing with the testing dataset. The target label for the damaged location is the debonding zone in the stiffened panel. Table 6 lists the division of debonding zones according to debonding location. Four types of testing were performed with the testing dataset, being 10, 15, 20 and 25 percent of the original dataset. The ML model used was a support vector machine (SVM). Figure 13 shows the performance of the SVM model in debonding zone identification at various train–test split ratios with the dataset transformed by DT models 1, 2 and 3, respectively. From the figure, it can be seen that SVM gives the best accuracy score for debonding zone identification when the FE data is transformed using DT Model 3.

Based on the two criteria and the respective performance levels of the DT models in fulfilling these criteria, it can be concluded that model 3 is the best choice to proceed with. Figure 12b also shows that the DT models with more trainable parameters perform better compared to the model with fewer trainable parameters. The visual comparison between FE-simulated and experimental vibration mode shape displacement data for debonding length 50 mm at 265 mm is shown in Figure 14a before transformation and Figure 14b after transformation using DT model 3. The figure clearly shows that the transformed FE-simulated data is very close to the experimental data.

Table 6. Debonding zone divisions.

Debonding Location (X)	105, 125, 145	165, 185, 205	225, 245, 265, 285, 305	325, 345, 365	385, 405, 425
Debonding Zone	Zone-1	Zone-2	Zone-3	Zone-4	Zone-5



(**a**) (**b**)

Figure 14. The first vibrational mode shape displacement data for scan points 1 to 208 (one-symmetric part): (**a**) original FE-simulated and experimental; (**b**) transformed FE-simulated and experimental data for debonding of 50 mm at 265 mm.

5.4. Damage Assessment

5.4.1. Feature Sensitivity Analysis

This section discusses the preprocessing of the transformed dataset and the debonding evaluation procedure. The data preprocessing procedure begins with feature-sensitive analysis, also known as feature extraction. The goal of feature extraction is to extract lower-dimensional features and more-susceptible-to-damage features from high-dimensional raw data. For the feature extraction process, a procedure based on Mahalanobis square distance (MSD) was adopted. MSD is a statistical measure used for assessing discrepancies between different datasets. Its use in damage detection is described in detail by Worden et al. [34]. The MSD of a multivariate dataset consists of *n* observations in p variables. The MSD is employed to estimate any observed data's discordance. Mathematically, for multivariate data, the MSD is defined as

$$MSD_{\zeta} = \left(X_{\zeta} - \underline{X}\right)^{T} S^{-1} \left(X_{\zeta} - \underline{X}\right) \tag{6}$$

where $\underline{X}$ is the sample mean of observation, X_{ζ} is an outlier, and S is the sample covariance matrix. In our exercise, the MSD estimation started with the creation of 2000 copies of an undamaged dataset by adding 0.2% of Gaussian noise to it. Then, MSD was estimated for each damaged case. It was observed that, for debonding locations away from the center location (i.e., away from location 265 mm), the MSD value was higher for the same debonding size. The MSD value was lowest for the debonding situated at the center location (at 265 mm). Also, it was observed that the MSD values were monotonically increasing with the size of debonding at the same debonding location. So, to set the threshold limit of the MSD value, it was required to study the MSD value for each debonding size located at the center (at 265 mm). An additional 100 copies of the dataset for each debonding size were created by adding 0.2 percent of Gaussian noise. In Figure 15, the estimated MSD value for the created dataset is shown for the corresponding debonding size, located at

265 mm. The mean difference between the MSD value for undamaged-specimen data and 15 mm of debonding gives a statistically significant result. Therefore, to set the threshold MSD value, the combination of 15 mm debonding at the 265 mm location was picked. Figure 16a displays the cumulative sum of the calculated MSD for each feature index number for the 15 mm debonding size in ascending order. The threshold MSD value is set as shown in the figure with the vertical line. Subsequently, the MSD values for all feature indices for each debonding case were calculated. Then, the calculated MSD was passed through the filter designed according to the threshold MSD value. The feature index number is preserved for values greater than the threshold MSD value. Finally, the set with the preserved index number is selected as the debonding sensitive feature vector. A total of 108 sets of feature indices passed the threshold MSD value. Hence the initial feature dimension of size 208 is reduced to 108 after this feature-sensitive analysis.

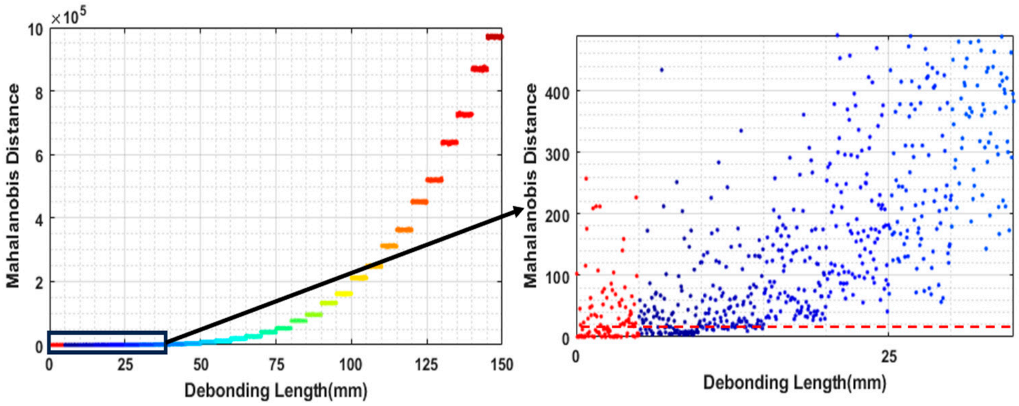
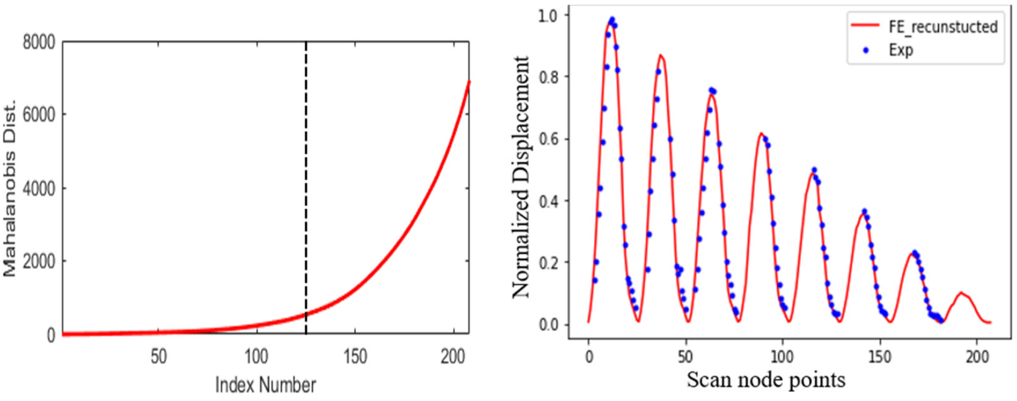


Figure 15. Estimated MSD values for debonding sizes of 10 mm to 150 mm located at 265 mm.



(**a**) (**b**)

Figure 16. (**a**) Threshold MSD value; (**b**) the filtered feature vector (blue dot).

5.4.2. Feature Dimension Reduction

The high-dimensional feature vector as an input to ML algorithms frequently results in overfitting and high computational costs, which limits the generalizability of the model. Keeping vital information concerning feature vectors in a lower-dimensional subspace aids in solving these problems. Among the many dimensional reduction techniques available, principal component analysis (PCA) is one that is widely used in ML, because it is the most basic and straightforward method of dimensionality reduction. In the present study, the first seven PCA components have been used, since they are able to cover 98% of the total variance in the dataset.

5.4.3. Dataset for Damaged Assessment

A total of 493 damaged cases were modelled in the FE simulation. There are 17 damaged locations, ranging from 105 mm to 425 mm along the stiffener length with a step size of 20 mm, and at each location, 29 debonding sizes, ranging from 10 mm to 150 mm with a step size of 5 mm. These FE datasets were transformed into the experimental domain using the DT model. The transformed FE datasets were used for the training and validation of ML models in damage assessment. Initially, all damage cases were labeled with their corresponding debonding location and debonding size. However, in the initial trial, it was observed that the prediction accuracy for debonding location and size were not reasonable. Therefore, the labels of debonding location and size were changed to debonding location zone and debonding size group. Table 7 lists the debonding location zones, and Table 8 lists the debonding size groups.

Table 7. Debonding location zones.

Debonding Location (X)	105, 125, 145	165, 185, 205	225, 245, 265, 285, 305	325, 345, 365	385, 405, 425
Debonding Zone	Zone-1	Zone-2	Zone-3	Zone-4	Zone-5

Table 8. Debonding size groups.

Debonding Size (mm)	**Size Group**
10, 15, 20, 25, 30	Group-1
35, 40, 45, 50, 55	Group-2
60, 65, 70, 75	Group-3
80, 85, 90, 95	Group-4
100, 105, 110, 115	Group-5
120, 125, 130, 135	Group-6
140, 145, 150	Group-7

5.4.4. Performance Evaluation of ML Models in Damage Assessment

The proposed methodology for assessing debonding includes location zone and size group prediction as a classification problem. Five ML algorithms, namely, SVM [35], Gradient Boosting (GB), Random Forest (RF) [36], Nearest Neighbors Classification (K-NN) [37] and Adaptive Boosting (ABC) were used as base learning models for a stacked model. The stacked model is an ensemble model that employs a two-stage training procedure. In the initial stage, the base learning model is trained, and in the subsequent stage, the outcome of the base model is used as an input feature for training the meta-model in order to produce the final prediction [38]. In general, the stacked model approach has been reported to provide better prediction accuracy, as compared to base-leaning models [39–43].

Two stacked models were created with the same base learner as mentioned above and logistic regression (LR) as a meta-learner (second level learner). One was for debonding location zone prediction and another for debonding size group prediction. The stacked models were trained with FE-transformed data, as input and location zone and size group as targets, respectively. The hyper-parameters of the base learners for the two stacked models were optimized using random grid search algorithms. In random grid search, a predefined range of values for each hyperparameter is specified, and the method picks random combinations of hyperparameters within those ranges for evaluation. The optimal set of hyperparameters is determined by training and evaluating the model for each set of hyperparameters and selecting the combination of parameters that results in the best performance.

Several performance matrices, including accuracy, sensitivity, specificity, F1 score, area under the curve (AUC) of the receiver operating characteristic (ROC) curve, and AUC of the precision recall curve, are commonly used to assess the performance of machine-learning models. These performance metrics, however, lack comprehensive information.

For instance, the AUC of the precision recall curve and the ROC curve are excessively broad, as they assess all decision criteria, even irrational ones. On the other hand, it is unfair to quantify metrics like accuracy, sensitivity, specificity, positive predictive value and F1 score at a single threshold that works well in certain situations but not others. To overcome these issues, Carrington et al., 2023 has proposed a novel performance evaluation technique called deepROC [44]. The deepROC evaluates performance across several projected risk categories. Deep ROC calculates the averages of sensitivity, specificity, positive and negative predictive values, and likelihood ratios (both positive and negative) for each group, as well as the group AUC and the normalized group AUC. In-depth information by group is provided by the deep ROC analysis, which often enhances model assessment over the regular ROC. For further information, readers are advised to read the work of Corrington et al. (2020) [44] and Corrington et al. (2023) [45]. In present study, we have employed the deepROC for the evaluation of the performance of ML models by estimating the AUC of the ROC.

5.4.5. Results with FE-Transformed Data

The entirety of the FE-transformed data was divided into a training dataset (85%) and a test dataset (15%). Then, the hyperparameters of each ML model were optimized with the training dataset using a random grid search algorithm with ten iterations and six cross-validations (CV). Once the hyperparameters of the ML models were tuned, the models were trained with training data and tested with the test datasets. Figure 17a,b show the confusion matrix for debonding location zone and size group predicted by the stacked model with test dataset. In the confusion matrix figure, the value in each square along the diagonal represents the number of true predicted labels and the value in the round bracket shows the fraction of true class labels predicted accurately for particular class labels. The figure shows that the prediction precision for a location zone varies from 0.92 to 0.62. For debonding size group prediction, the precision varied from 0.27 to 0.86. The overall performance of ML models in terms of the AUC of DeepROC is listed in Table 9. The values in the table help in reiterating the conclusion that the stacked model is better, as compared to the base ML models, in both debonding zone localization and size group prediction. It also shows again that location zone is better-predicted than location size group. Overall, the ML models give a reasonable accuracy in both debonding location zone and size group predictions with FE-transformed data.

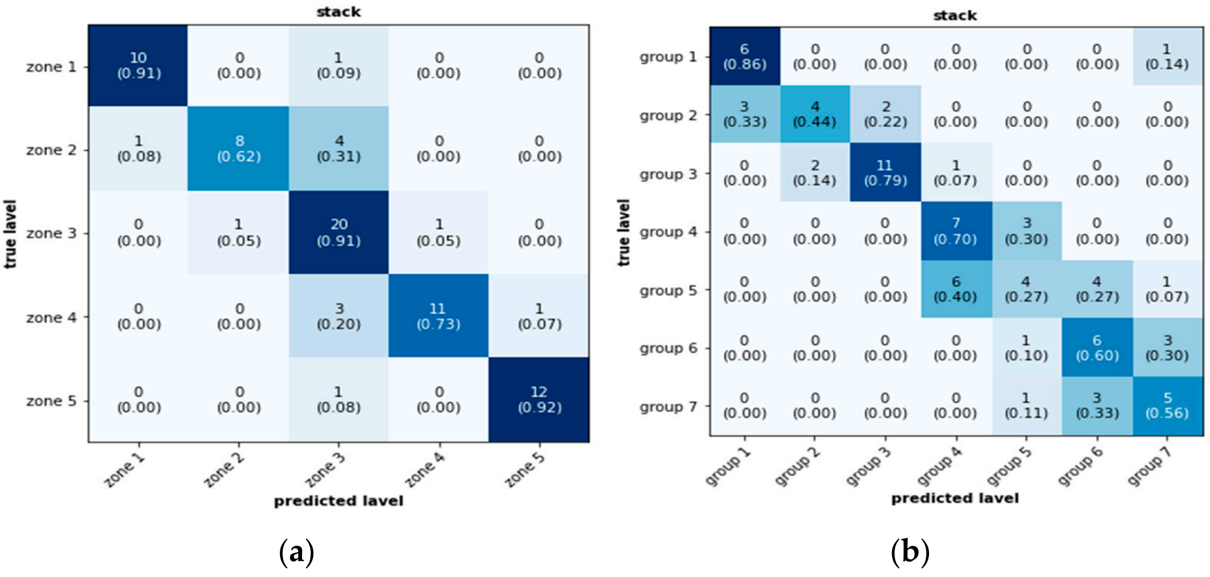


Figure 17. Confusion matrix for (**a**) debonding location zone and (**b**) debonding size group, as predicted by stacked model.

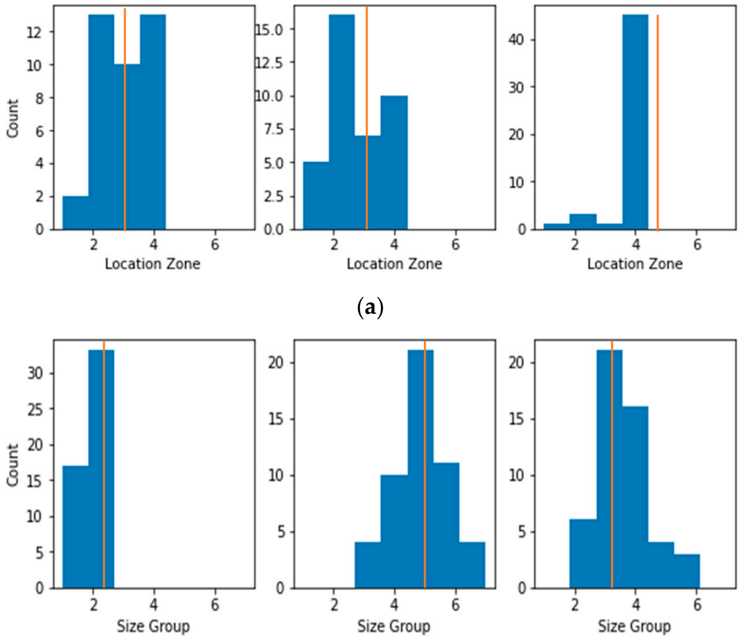
Table 9. AUC for DeepROC for ML models.

ML Model	Location AUC (DeepROC)	Size AUC (DeepROC)
SVM	0.717	0.720
RF	0.785	0.905
ABC	0.549	0.6823
GBC	0.588	0.907
k-nn	0.632	0.887
Stack	0.920	0.952

SVM, Support Vector Machine; RF, Random Forest; ABC, Adaptive Boosting, GBC, Gradient Boosting, K-NN, k-nearest neighbor; Stack, Stacked model.

5.4.6. Results with Experimental Data

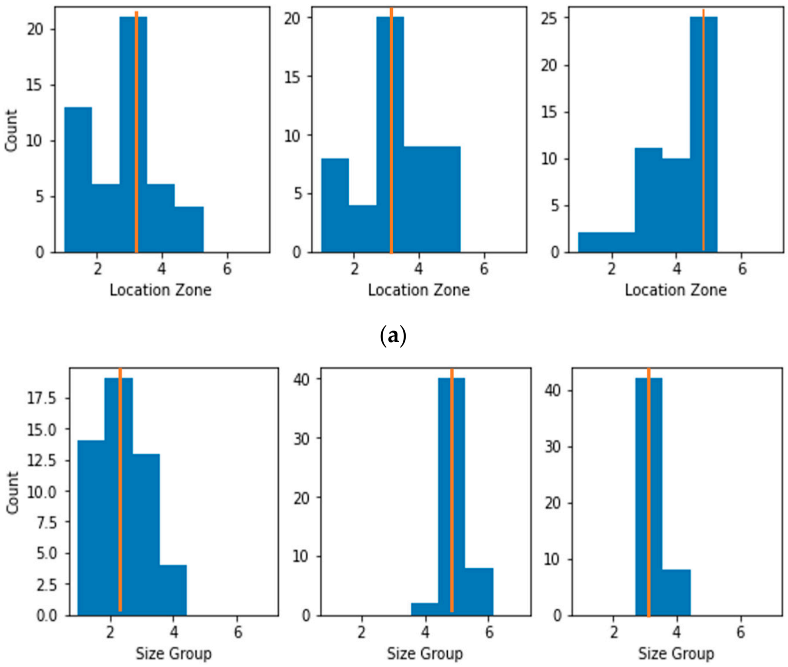
After obtaining satisfactory results in debonding location zones and size group predictions with FE-transformed data using the ML model, the next task was to test the efficacy of the methodology with experimental data. Only the stacked model was considered for the experimental dataset. The two stacked models which were trained with FE-transformed data for the debonding location zone and size group dataset were tested with the experimental datasets for location zone and size group prediction, respectively. There were three damaged experimental specimens. Fifty experimental datasets were generated for each of them. These datasets were used as inputs for the trained stacked models. Figure 18a,b show the prediction results for debonding location zone and size group, respectively. The x-axes of the figures are the damage labels, the y-axes are the counts (i.e., the number of times a particular damage label has been predicted), and the yellow lines in the figure show the actual damage labels. From these figures, it can be seen that, for the debonding zone location and size group, the predicted labels are populated around the actual class label. However, for debonding size groups, the maximum counts of predicted and actual labels are the same, which is not the case for location zone prediction. However, for the stacked model trained with FE-transformed data with two-percent added Gaussian noise, the prediction accuracy with experimental data considerably improved for both debonding size group and debonding location zone, as shown in Figure 19.



(a)

(b)

Figure 18. Results with experimental data, with stacked model trained with FE-transformed data without noise for debonding (**a**) location zone and (**b**) size group; the histogram shows predicted labels, while the actual labels are marked by a yellow line.



(**a**)

(**b**)

Figure 19. Results with experimental data, with stacked model trained with FE-transformed data with two-percent Gaussian noise for debonding (**a**) location zone and (**b**) size group; the histogram shows predicted labels, while the actual labels are marked by a yellow line.

6. Conclusions and Future Work

This work presents a new method of generating an FE database for SHM which is very close to an experimental database. The method used is a deep-learning-based domain-transfer model which transforms the FE domain data into experimental data and minimizes the differences between them. The methodology is implemented for debonding localization and size estimation for metallic stiffened panels for vibration-based damage assessment. A bootstrapping technique was used to generate sufficient experimental data from a few samples of data collected from experiments using an LDV set-up. For FE-simulated data, a sufficient number of copies of data were created by adding noise to individual instances. The FE-simulated data was used as the input and experimental data was used as the target in the DL-based DT model. This pretrained DT model (with undamaged-specimen experimental and FE-simulated databases) was used to transform the FE-simulated damaged-specimen database to the experimental domain. Three DT models were designed, and a comparative study showed that a stacked DT model with a larger number of trainable parameters transformed the data into the experimental domain more effectively. This transformed FE dataset was used to train and evaluate ML models. Then, the trained ML model was tested with the experimentally obtained damaged-specimen dataset. The results are satisfactory for predicting the debonding location zones and size groups in stiffened panels.

The key contribution of this work is the establishment of a method of creation of a domain-transfer model for generation of large databases from finite-element-based models which are very close to actual experimental results in their statistical distribution. Though this has been demonstrated with the specific example of debonding locations and size predictions of stiffened panels using vibration signals, the technique has a much wider scope of application. It can be extended to other categories of structural-health monitoring problems and even to other domains encountering the shared difficulty of generating a large number of experimental data points for training machine-learning models.

Author Contributions: A.K.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft; A.G.: conceptualization, methodology, investigation, resources, supervision, writing—review and editing; S.B.: experimental resources, investigation, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and/or analyzed within the current study are available from the corresponding author on reasonable request.

Acknowledgments: The authors acknowledge the support provided by LDV lab, Aerospace Department, IIT Bombay, during the experimental work.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. The architectural details of the DT models.

Model-1	
Layer (Type)	**Output Shape**
Input	(208)
Linear-1 (Tanh)	(1024)
Linear-2 (Tanh)	(512)
Output	(208)
Total parameters	845,520
Model-2	
Layer (Type)	Output Shape
Input	(208)
Conv1D-1 (Tanh) (BN)	(16, 103)
Conv1D-2 (Tanh) (BN)	(32, 50)
Conv1D-3 (Tanh) (BN)	(64, 24)
Conv1D-4 (Tanh) (BN)	(128, 11)
Linear-1 (Tanh)	(1024)
Linear-2 (Tanh)	(512)
Output	(208)
Total parameters	2,118,112
Model-3	
Layer (type)	Output Shape
Input	(208)
Conv1D-1 (Tanh) (BN)	(16, 103)
Conv1D-2 (Tanh) (BN)	(32, 50)
Conv1D-3 (Tanh) (BN)	(64, 24)
Conv1D-4 (Tanh) (BN)	(128, 11)
Linear-1 (Tanh)	(1024)
Linear-2 (Tanh)	(1408)
TransConv1D-1 (Tanh) (BN)	(64, 24)
TransConv1D-2 (Tanh) (BN)	(32, 50)
TransConv1D-3 (Tanh) (BN)	(16, 102)
Output	(208)
Total parameters	2,905,641

BN = batch normalization, Conv1D = 1D-convolution channel.

References

1. Farrar, C.R.; Worden, K.; Wiley, J. *Structural Health Monitoring: A Machine Learning Perspective*; John Wiley and Sons: Hoboken, NJ, USA, 2012.
2. Figueiredo, E.; Park, G.; Farrar, C.R.; Worden, K.; Figueiras, J. Machine learning algorithms for damage detection under operational and environmental variability. *Struct. Health Monit.* **2010**, *10*, 559–572. [CrossRef]
3. Yang, Z.; Yang, H.; Tian, T.; Deng, D.; Hu, M.; Ma, J.; Gao, D.; Zhang, J.; Ma, S.; Yang, L.; et al. A review on guided-ultrasonic-wave-based structural health monitoring: From fundamental theory to machine learning techniques. *Ultrasonics* **2023**, *133*, 107014. [CrossRef] [PubMed]
4. Gordan, M.; Sabbagh-Yazdi, S.-R.; Ismail, Z.; Ghaedi, K.; Carroll, P.; McCrum, D.; Samali, B. State-of-the-art review on advancements of data mining in structural health monitoring. *Measurement* **2022**, *193*, 110939. [CrossRef]
5. Toh, G.; Park, J. Review of Vibration-Based Structural Health Monitoring Using Deep Learning. *Appl. Sci.* **2020**, *10*, 1680. [CrossRef]
6. Azimi, M.; Eslamlou, A.D.; Pekcan, G. Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review. *Sensors* **2020**, *20*, 2778. [CrossRef] [PubMed]
7. Jayawickrema, U.; Herath, H.; Hettiarachchi, N.; Sooriyaarachchi, H.; Epaarachchi, J. Fibre-optic sensor and deep learning-based structural health monitoring systems for civil structures: A review. *Measurement* **2022**, *199*, 111543. [CrossRef]
8. Civera, M.; Surace, C. Non-Destructive Techniques for the Condition and Structural Health Monitoring of Wind Turbines: A Literature Review of the Last 20 Years. *Sensors* **2022**, *22*, 1627. [CrossRef]
9. Zhang, C.; Mousavi, A.A.; Masri, S.F.; Gholipour, G.; Yan, K.; Li, X. Vibration feature extraction using signal processing techniques for structural health monitoring: A review. *Mech. Syst. Signal Process.* **2022**, *177*, 109175. [CrossRef]
10. Eltouny, K.; Gomaa, M.; Liang, X. Unsupervised Learning Methods for Data-Driven Vibration-Based Structural Health Monitoring: A Review. *Sensors* **2023**, *23*, 3290. [CrossRef]
11. Kudva, J.N.; Munir, N.; Tan, P.W. Damage detection in smart structures using neural networks and finite-element analyses. *Smart Mater. Struct.* **1992**, *1*, 108–112. [CrossRef]
12. Worden, K.; Ball, A.D.; Tomlinson, G.R. Fault location in a framework structure using neural networks. *Smart Mater. Struct.* **1993**, *2*, 189–200. [CrossRef]
13. Sbarufatti, C.; Manes, A.; Giglio, M. Performance optimization of a diagnostic system based upon a simulated strain field for fatigue damage characterization. *Mech. Syst. Signal Process.* **2013**, *40*, 667–690. [CrossRef]
14. Sbarufatti, C.; Manson, G.; Worden, K. A numerically-enhanced machine learning approach to damage diagnosis using a Lamb wave sensing network. *J. Sound Vib.* **2014**, *333*, 4499–4525. [CrossRef]
15. Su, Z.; Ye, L. Lamb wave-based quantitative identification of delamination in CF/EP composite structures using artificial neural algorithm. *Compos. Struct.* **2004**, *66*, 627–637. [CrossRef]
16. Su, Z.; Ye, L. Lamb Wave Propagation-based Damage Identification for Quasi-isotropic CF/EP Composite Laminates Using Artificial Neural Algorithm: Part I—Methodology and Database Development. *J. Intell. Mater. Syst. Struct.* **2005**, *16*, 97–111. [CrossRef]
17. Gardner, P.; Liu, X.; Worden, K. On the application of domain adaptation in structural health monitoring. *Mech. Syst. Signal Process.* **2020**, *138*, 106550. [CrossRef]
18. Zhang, S.; Li, C.M.; Yang, J.; Ye, W. Effective combination of modeling and experimental data with deep metric learning for guided wave-based damage localization in plates. *Mech. Syst. Signal Process.* **2022**, *172*, 108979. [CrossRef]
19. De Fenza, A.; Sorrentino, A.; Vitiello, P. Application of Artificial Neural Networks and Probability Ellipse methods for damage detection using Lamb waves. *Compos. Struct.* **2015**, *133*, 390–403. [CrossRef]
20. Kumar, A.; Guha, A.; Banerjee, S. A Methodology for Diagnosis of Damage by Machine Learning Algorithm on Experimental Data. In *Lecture Notes in Civil Engineering*; Rizzo, P., Milazzo, A., Eds.; Springer: Cham, Switzerland, 2021; Volume 128, pp. 91–105. [CrossRef]
21. Barthorpe, R.; Manson, G.; Worden, K. On multi-site damage identification using single-site training data. *J. Sound Vib.* **2017**, *409*, 43–64. [CrossRef]
22. Bao, N.; Zhang, T.; Huang, R.; Biswal, S.; Su, J.; Wang, Y. A Deep Transfer Learning Network for Structural Condition Identification with Limited Real-World Training Data. *Struct. Control Health Monit.* **2023**, *2023*, 8899806. [CrossRef]
23. Efron, B.; Tibshirani, R. The Bootstrap Method for Assessing Statistical Accuracy. *Behaviormetrika* **1985**, *12*, 1–35. [CrossRef]
24. Semaan, R. The uncertainty of the experimentally-measured momentum coefficient. *Exp. Fluids* **2020**, *61*, 248. [CrossRef]
25. Carroll, A.; Sanders-O'connor, E.; Forrest, K.; Fynes-Clinton, S.; York, A.; Ziaei, M.; Flynn, L.; Bower, J.M.; Reutens, D. Improving Emotion Regulation, Well-being, and Neuro-cognitive Functioning in Teachers: A Matched Controlled Study Comparing the Mindfulness-Based Stress Reduction and Health Enhancement Programs. *Mindfulness* **2021**, *13*, 123–144. [CrossRef]
26. Mahmoud, O.; Dudbridge, F.; Smith, G.D.; Munafo, M.; Tilling, K. A robust method for collider bias correction in conditional genome-wide association studies. *Nat. Commun.* **2022**, *13*, 619. [CrossRef]
27. Renault, L.; McWilliams, J.C.; Kessouri, F.; Jousse, A.; Frenzel, H.; Chen, R.; Deutsch, C. Evaluation of high-resolution atmospheric and oceanic simulations of the California Current System. *Prog. Oceanogr.* **2021**, *195*, 102564. [CrossRef]
28. Ince, T.; Kiranyaz, S.; Eren, L.; Askar, M.; Gabbouj, M. Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks. *IEEE Trans. Ind. Electron.* **2016**, *63*, 7067–7075. [CrossRef]

29. Zan, T.; Wang, H.; Wang, M.; Liu, Z.; Gao, X. Application of Multi-Dimension Input Convolutional Neural Network in Fault Diagnosis of Rolling Bearings. *Appl. Sci.* **2019**, *9*, 2690. [CrossRef]
30. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [CrossRef]
31. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
32. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 2nd International Conference on Machine Learning (ICML 2015), Lille, France, 6–11 July 2015; Volume 1, pp. 448–456.
33. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
34. Worden, K.; Manson, G.; Fieller, N. Damage detection using outlier analysis. *J. Sound Vib.* **2000**, *229*, 647–667. [CrossRef]
35. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
36. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
38. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. [CrossRef]
39. Aldave, R.; Dussault, J.P. Systematic Ensemble Learning for Regression. *arXiv* **2014**, arXiv: 1403.7267. [CrossRef]
40. Cao, C.; Wang, Z. IMCStacking: Cost-sensitive stacking learning with feature inverse mapping for imbalanced problems. *Knowledge-Based Syst.* **2018**, *150*, 27–37. [CrossRef]
41. Kumar, A.; Guha, A.; Banerjee, S. Improving Prediction Accuracy for Debonding Quantification in Stiffened Plate by Meta-Learning Model. In Proceedings of the International Conference on Big Data, Machine Learning and Their Applications, Silchar, India, 19–20 December 2021; pp. 51–63. [CrossRef]
42. Zhai, B.; Chen, J. Development of a stacked ensemble model for forecasting and analyzing daily average PM2.5 concentrations in Beijing, China. *Sci. Total Environ.* **2018**, *635*, 644–658. [CrossRef]
43. Ozay, M.; Yarman-Vural, F.T. Hierarchical distance learning by stacking nearest neighbor classifiers. *Inf. Fusion* **2016**, *29*, 14–31. [CrossRef]
44. Carrington, A.M.; Manuel, D.G.; Fieguth, P.W.; Ramsay, T.; Osmani, V.; Wernly, B.; Bennett, C.; Hawken, S.; Magwood, O.; Sheikh, Y.; et al. Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 329–341. [CrossRef]
45. Carrington, A.M.; Fieguth, P.W.; Qazi, H.; Holzinger, A.; Chen, H.H.; Mayr, F.; Manuel, D.G. A new concordant partial AUC and partial c statistic for imbalanced data in the evaluation of machine learning algorithms. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 4. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.