

Research Article

A Damage Detection Method Using Neural Network Optimized by Multiple Particle Collision Algorithm

Sergio V. Farias ¹, Osamu Saotome ¹, Haroldo F. Campos Velho ²,
and Elcio H. Shiguemori ³

¹Aeronautics Institute of Technology (ITA), São José dos Campos 12228-900, Brazil

²National Institute for Space Research (INPE), São José dos Campos 12227-010, Brazil

³Institute of Advanced Studies (IEAv), São José dos Campos 12228-001, Brazil

Correspondence should be addressed to Sergio V. Farias; sertone@gmail.com

Received 22 March 2021; Revised 11 June 2021; Accepted 13 July 2021; Published 6 August 2021

Academic Editor: Stelios M. Potirakis

Copyright © 2021 Sergio V. Farias et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A critical task of structural health monitoring is damage detection and localization. Lamb wave propagation methods have been successfully applied for damage identification in plate-like structures. However, Lamb wave processing is still a challenging task due to its multimodal and dispersive characteristics. To address this issue, data-driven machine learning approaches as artificial neural network (ANN) have been proposed. However, the effectiveness of ANN can be improved based on its architecture and the learning strategy employed to train it. The present paper proposes a Multiple Particle Collision Algorithm (MPCA) to design an optimum ANN architecture to detect and locate damages in plate-like structures. For the first time in the literature, the MPCA is applied to find damages in plate-like structures. The present work uses one piezoelectric transducer to generate Lamb wave signals on an aluminum plate structure and a linear array of four transducers to capture the scattered signals. The continuous wavelet transform (CWT) processes the captured signals to estimate the time-of-flight (ToF) that is the ANN inputs. The ANN output is the damage spatial coordinates. In addition to MPCA optimization, this paper uses a quantitative entropy-based criterion to find the best mother wavelet and the scale values. The presented experimental results show that MPCA is capable of finding a simple ANN architecture with good generalization performance in the proposed damage localization application. The proposed method is compared with the 1-dimensional convolutional neural network (1D-CNN). A discussion about the advantages and limitations of the proposed method is presented.

1. Introduction

In safety-critical systems, failure detection and prognostic (FDP) approaches are essential to avoid catastrophic failures. The two main tasks of FDP methods are incipient failure detection and remaining useful life estimation through prognostics techniques [1]. The FDP approaches can be applied in different critical systems as wind turbine [2], gas turbines [3], power systems [4, 5], and transmission lines [6].

Structural health monitoring (SHM) investigates the damage detection and prognostic in structural components of critical systems such as aircraft and bridges. The SHM system provides an online solution for performing in-service

structural analysis reducing maintenance costs [7]. The SHM approaches have been developed for mechanical, aerospace, and civil engineering applications [8]. The most important task of the SHM strategy is the damage identification procedure that can be classified into four levels: detection, location, quantification, and prediction [9].

SHM systems perform structural inspection in real time using embedded sensors [10]. A usual sensor and actuator option is piezoelectric (PZT) transducers [11–15]. The advantages of PZT transducers are small size, light weight, and low power consumption [16].

Predominant PZT-based SHM methods are vibration, electromechanical impedance, and guided wave [12, 17].

The vibration methods compare vibration signatures of damaged and undamaged structures. Usually, a vibration signature is obtained through modal analysis [8]. The electromechanical impedance methods are based on the electromechanical impedance signature [12]. The guided wave methods are based on propagation properties of the acoustic wave as Lamb waves. The Lamb waves are guided waves that propagate in thin-walled structures, and they can propagate long distances with low attenuation and have high sensibility to detect small damages [18, 19]. Therefore, Lamb waves have been a promising tool for damage detection and localization in plate-like structures.

The SHM methods can be divided into two main approaches: physical-based or data-driven [20]. The physical-based approaches require a detailed model of the structure while the data-driven approaches are based on database analysis. Phased-array and directional filter [21–23], subspace approach [24–26], time reversal [27, 28], and ellipse-based image [16, 29, 30] are methodologies based on the physical model of Lamb wave propagation. It is a challenging task due to its multimodal and dispersive characteristics [18, 27].

The data-driven approaches are machine learning and deep learning [8, 31, 32]. Machine learning techniques are data-driven approaches that learn patterns found in the database and can be classified into supervised learning and unsupervised learning [27, 33]. The most common supervised learning methods are the artificial neural network and support vector machine [33]. Some advantages of these techniques are low computational cost and high generalization capability. Deep learning is a subbranch of machine learning, and it has the capability to deal with a large dataset [20, 33]. Recent studies have proposed the 1-dimensional convolutional neural network (1D-CNN) for structural damage detection and localization [8, 10, 13, 20]. The main advantage of 1D-CNN is automatic feature extraction performed through its initial convolutional layers [10, 13, 34]. However, CNN has a high computational cost, and its architecture design is a difficult task [35].

Several strategies using ANN for damage localization and classification in plate-like structures have been proposed. Lu et al. [36] implemented an inverse method based on the feed-forward artificial neural network for damage identification using PZT transducers. The ANN configuration (the number of hidden layers and the number of neurons in each layer) was determined using a rule of thumb. The discrete wavelet transform was applied to signal denoising before feature extraction. Yelve and Mulla [37] proposed damage detection in an aluminum plate using ANN and PZT transducers. The ANN inputs were damage indexes obtained from twelve actuator sensor paths, and the specific software selected the ANN architecture automatically. Feng et al. [30] presented two methods for damage detection in anisotropic woven-fabric carbon fiber reinforced polymer plate using PZT transducers. The first method was a probabilistic approach by constructing a probability matrix, and the second method was an ANN. The ANN inputs were the time-of-flight of the wave acquired by PZT transducers, and a try-and-error method selected the ANN configuration. Hesser et al. [38] proposed an ANN and support vector machine to locate

low impact in the aluminum plate. An ANN with one hidden layer was proposed, and the number of neurons was found through a try-and-error method. However, these approaches do not use any method to find an optimized ANN topology. The best choice of neural network topology, weight values, and activation function for a particular application is a difficult task. ANN topology has a high impact on its performance. A small number of neurons may reduce the ANN learning capacity, but an excessive number of neurons may reduce its generalization capacity [39].

In this context, the metaheuristic approach formulates the ANN parameter identification task into an optimization problem. The most common method is ANN weight optimization where the metaheuristic algorithm searches for the weight values that minimize a cost function. This approach focuses on the ANN training process where the metaheuristic algorithm provides a way to escape from local minimums, which is the main problem in standard gradient-based methods. Several metaheuristic algorithms have been proposed to improve the ANN training process, such as genetic algorithm [40, 41], particle swarm optimization [42], and grasshopper optimization [43]. However, to improve the ANN performance, the cost function needs to balance the weight values, ANN topology, and learning parameters. The genetic algorithm is the common metaheuristic method of ANN architecture optimization [44–47].

The Multiple Particle Collision Algorithm (MPCA) is a metaheuristic that minimizes a mono-objective function providing the simplest neural network topology (lowest number of neurons and faster convergence during the training phase) with the best performance (lowest error) [48]. The MPCA optimization algorithm takes into account all relevant ANN parameters as the number of hidden layers, the number of neurons in each hidden layer, the weight values, the learning rate, the momentum constant, and the activation function [49]. MPCA has been successfully applied to climate prediction applications [39]. The present paper proposes an optimized ANN for damage detection in a plate-like structure using the MPCA. For the first time in the literature, the ANN optimized by MPCA is applied to find damages in plate-like structures. The ANN inputs are the time-of-flight of the Lamb wave reflection captured by four piezoelectric transducers, and the ANN output is the damage localization.

In the damage localization methods using ANN, a common approach is to use four PZT transducers placed in the corner of the inspected plate region [30, 36, 37]. The damage position is estimated using the time-of-flight information. This configuration is similar to a pitch-catch method finding damages in the delimited area by the transducers. In the present paper, a linear array of PZT transducers is investigated. The transducer placement is similar to a phased-array configuration, so the damage location information is strongly related to the time-of-flight and the time delay between adjacent transducers [18]. Using a linear array of transducers is possible to inspect a large area of the plate.

In the proposed damage localization method, due to the importance of the ToF estimation, the continuous wavelet transform is implemented to improve the arrival time measurement of the scattered wave from damage. The mother

wavelet is the main CWT parameter, as it may significantly influence the performance of the transformation. Recent works have proposed wavelet transform for damage detection and localization [4, 6, 30]. However, these works do not investigate a quantitative approach for mother wavelet selection. The present work uses a quantitative criterion, based on the Shannon entropy calculation, to select the best mother wavelet function [50]. The Shannon entropy criterion selects the most relevant scale values, reducing the CWT calculation.

The proposed new method combines two optimization approaches: optimized ANN by MPCA and optimized CWT by entropy criterion. The experimental verification of the proposed method is performed on an aluminum plate structure. A mass of lead is employed to simulate several damage scenarios, providing a database for training the ANN and 1D-CNN. The performance of the proposed method is compared with two 1D-CNN. The first 1D-CNN uses as input the ToF features extracted using CWT, and the second 1D-CNN performs feature extraction directly from raw data acquired from PZT transducers. A discussion about the advantages and limitations of the proposed method is presented.

2. Background

2.1. Continuous Wavelet Transform. Continuous wavelet transform CWT is a linear transformation that decomposes the input signal $x(t)$ over the scaled and translated versions of the mother wavelet $\psi(t)$, as shown by the following equation [51]:

$$\text{CWT}(s, \tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt, \quad (1)$$

where s is the scale, τ is the translation parameter, and $(*)$ indicates the complex conjugate of mother wavelet function $\psi(t)$. Using a proper scale-to-frequency relationship, CWT provides a time-frequency analysis similar to the short-time Fourier transform. However, CWT is more effective in representing nonstationary signals in the time-frequency domain [52, 53].

A challenge in using wavelet transform is to establish a criterion to select the mother wavelet. A usual qualitative approach is a method based on a similarity between the mother wavelet and the analyzed signal.

A suitable quantitative approach is Shannon entropy-based optimization. This criterion consists of calculating the normalized Shannon entropy of the CWT coefficients as defined by [50]:

$$\text{Sh}(s) = \sum_{n=1}^N \left(\frac{\text{CWT}(s, n \cdot T_s)}{A} \cdot \ln \left(\frac{\text{CWT}(s, n \cdot T_s)}{A} \right) \right), \quad (2)$$

$$A = \sum_{n=1}^N \sum_{a=1}^M \text{CWT}(s(a), n \cdot T_s), \quad (3)$$

where s is the scale vector with M elements, T_s is the sample time, and N is the number of samples of the $x(t)$ signal. Shannon entropy measures the CWT randomness in the time-scale domain, where a low entropy value indicates high energy concentration. Therefore, the best mother wavelet has the lowest value of Shannon entropy (Sh).

2.2. Hilbert Transform. The Hilbert transform $F_{\text{Hi}}(t)$ of the real signal $x(t)$ can be represented by the equation below [54]:

$$F_{\text{Hi}}(t) = \frac{-1}{\pi t} * x(t). \quad (4)$$

The Fourier transform of the $F_{\text{Hi}}(t)$ signal is expressed as

$$F_{\text{Hi}}(\omega) = (i \cdot \text{sgn}(\omega)) \cdot X(\omega), \quad (5)$$

where $\text{sgn}(\omega)$ is the sign function, $X(\omega)$ is the Fourier transform of input signal $x(t)$, and i is the imaginary number. According to Equation (5), the Hilbert transform is a filter in which the amplitudes of spectral components of the input signal $x(t)$ are left unchanged, but their phases are shifted by 90 degrees (see [54] page 359). This property can be used to calculate the instantaneous amplitude, or envelope, of signal $x(t)$. The $R(t)$ envelope of signal $x(t)$ is given by [52, 54]

$$R(t) = \sqrt{x(t)^2 + F_{\text{Hi}}(t)^2}. \quad (6)$$

2.3. Artificial Neural Networks. Artificial neural networks are computational methods that emulate the human brain learning process. The simple processing units, known as artificial neurons, compose the ANN and are organized through several layers. Typically, the ANN has one input layer, one or more hidden layers, and one output layer. The synaptic weights interconnect the neurons and store the ANN knowledge [39, 55].

Multilayer Perceptron Neural Network (MLP-NN) is a feed-forward model used mainly as a classifier, associative memory, or regression. Figure 1 shows an MLP neural network with just one hidden layer, where

- (i) $\vec{z} = (z_1, z_2, \dots, z_M)$ is the ANN input vector
- (ii) $\vec{x} = (x_1, x_2, \dots, x_N)$ is the ANN hidden layer vector
- (iii) $\vec{y} = (y_1, y_2, \dots, y_K)$ is the ANN output vector

Equations (7) and (8) describe the neuron x_n of the hidden layer and the neuron y_k of the output layer, respectively,

$$x_n = g_1 \left(\sum_{m=1}^M v_{nm} \cdot z_m + b_n^x \right), \quad (7)$$

$$y_k = g_2 \left(\sum_{n=1}^N \omega_{kn} \cdot x_n + b_k^y \right), \quad (8)$$

being b_{β}^{α} ($\alpha = x, y$ and $\beta = n, m$) the bias, v_{nm} and ω_{kn} the weight matrices of the connection between the neuron x_n and the neuron z_m , and y_k with the neuron x_n . The neuron

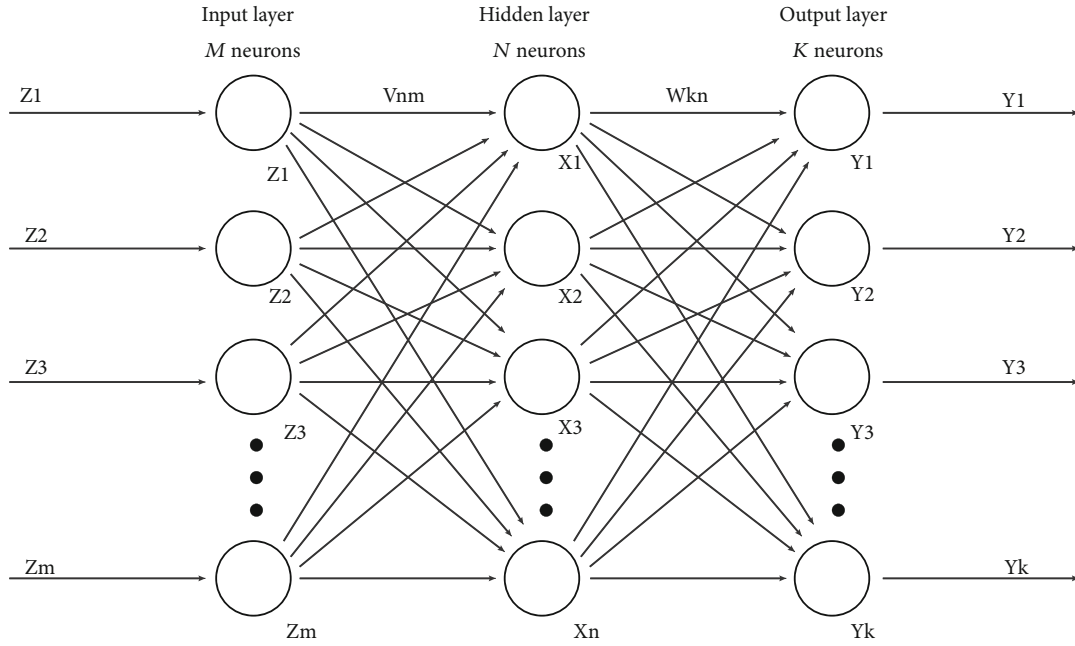


FIGURE 1: MLP neural network.

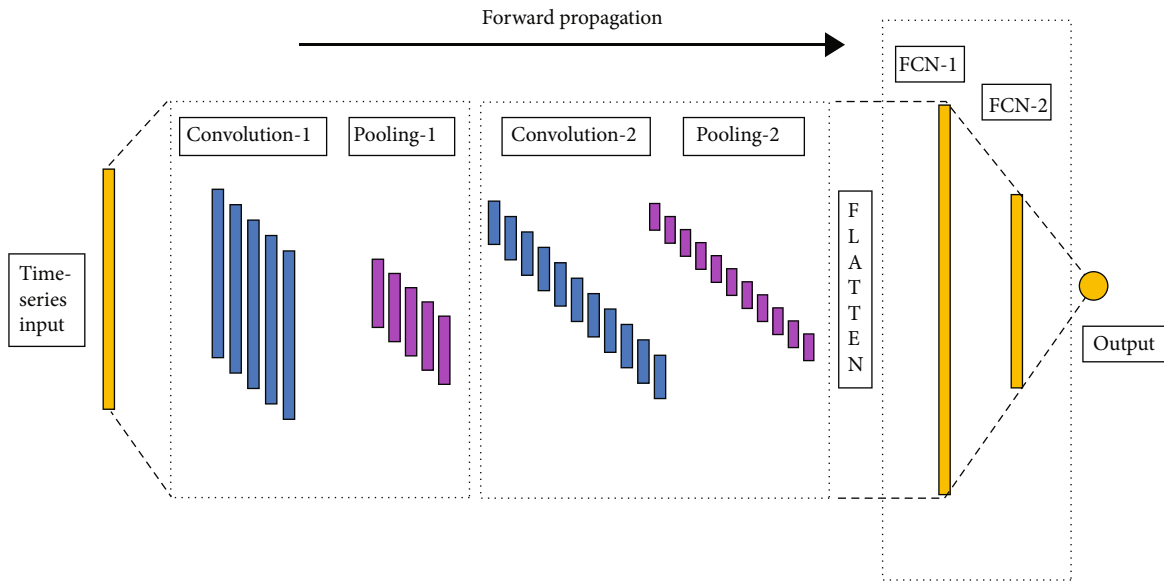


FIGURE 2: 1D convolutional neural network.

activation function $g_{1,2}(\cdot)$ is a nonlinear and smooth function [55]. The frequently used activation functions are sigmoid and hyperbolic tangent.

ANN training is the neuron's weight adjustment process and can be supervised or unsupervised. For an MLP-NN, the gradient-based and stochastic-based approaches are the main supervised training methods [43]. Among gradient-based methods, the frequently used is back-propagation. The back-propagation algorithm calculates the new weights using two steps: forward and backward [39]. In the first step, the input data is applied to the MLP-NN input, starting the data processing. The MLP-NN forward calculations propagate the

results layer-by-layer producing the MLP-NN output. The network output y_k is subtracted from an expected output of d_k to produce an error value [55]:

$$E_{\text{train}} = \sum_{k=1}^K (d_k - y_k)^2. \quad (9)$$

After that, this error is backward propagated from the MLP-NN output to the input. Finally, the neuron weights are adjusted using the delta rule for minimizing the calculated error.

```

1: procedure MPCAOptimization (integer  $N_p$ )
2:   Generate an initial solution: Old-Config
3:   Best-Fitness = Fitness(Old-Config)
4:   Update Blackboard
5:   for  $n = 1 \rightarrow N_p$  particles do
6:     for  $m = 1 \rightarrow M_i$  iterations do
7:       Update Blackboard
8:       Perturbation()
9:       if Fitness(New-Config) > Fitness(Old-Config) then
10:        if Fitness(New-Config) > Best-Fitness then
11:          Best-Fitness = Fitness(New-Config)
12:        end if
13:        Old-Config = New-Config
14:        Exploration()
15:      else
16:        Scattering()
17:      end if
18:    end for
19:  end for
20:  return New-Config
21: end procedure

```

ALGORITHM 1: Pseudocode for MPCA.

The delta rule can be expressed as [55]:

$$\Delta v_{nm} = -\eta \frac{\partial E_{\text{train}}}{\partial v_{nm}} \quad (n = 1, 2, \dots, N, m = 1, 2, \dots, M), \quad (10)$$

$$\Delta \omega_{kn} = -\eta \frac{\partial E_{\text{train}}}{\partial \omega_{kn}} \quad (k = 1, 2, \dots, K, n = 1, 2, \dots, N), \quad (11)$$

where Δv_{nm} and $\Delta \omega_{kn}$ are the weight correction factors and η is the learning rate parameter.

The feed-forward neural networks are popular mainly because of their generalization capability. The generalization refers to the ANN ability to correctly process data that is not part of training data. Therefore, the generalization performance can be estimated by

$$E_{\text{gen}} = \sum_{p=1}^P (d_p - y_p)^2. \quad (12)$$

The P data elements are not part of the K data elements—those ones were used to train the ANN [49].

2.4. 1D Convolutional Neural Networks. Convolutional neural networks (CNN) were inspired by the mammalian nervous system. The usual applications of CNN are regression and classification problems [34]. Typically, a CNN is composed of three types of processing layers: convolutional layers, pooling layers, and fully connected neural network (FCN) layers. Figure 2 shows an example of 1D-CNN.

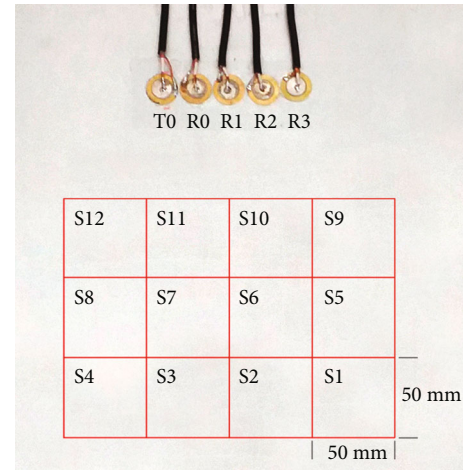


FIGURE 3: Aluminum plate representation.

A convolutional layer extracts relevant features from the input data. Equation (13) describes the convolution operation between layers l and $l-1$:

$$y_k^l = f \left(b_k^l + \sum_{i=1}^{N_{l-1}} \text{Conv1D}(\omega_{ik}^{l-1}, s_i^{l-1}) \right), \quad (13)$$

where y_k^l is the output, b_k^l is the bias of the k^{th} neuron at layer l , s_i^{l-1} is the output of the i^{th} neuron at layer $l-1$, ω_{ik}^{l-1} is the kernel from the i^{th} neuron at layer $l-1$ to the k^{th} at layer l , and $\text{Conv1D}(\cdot)$ is the 1D discrete convolution without zero padding [34]. The pooling layer performs a downsampling operation resulting in a dimensional reduction. Pooling

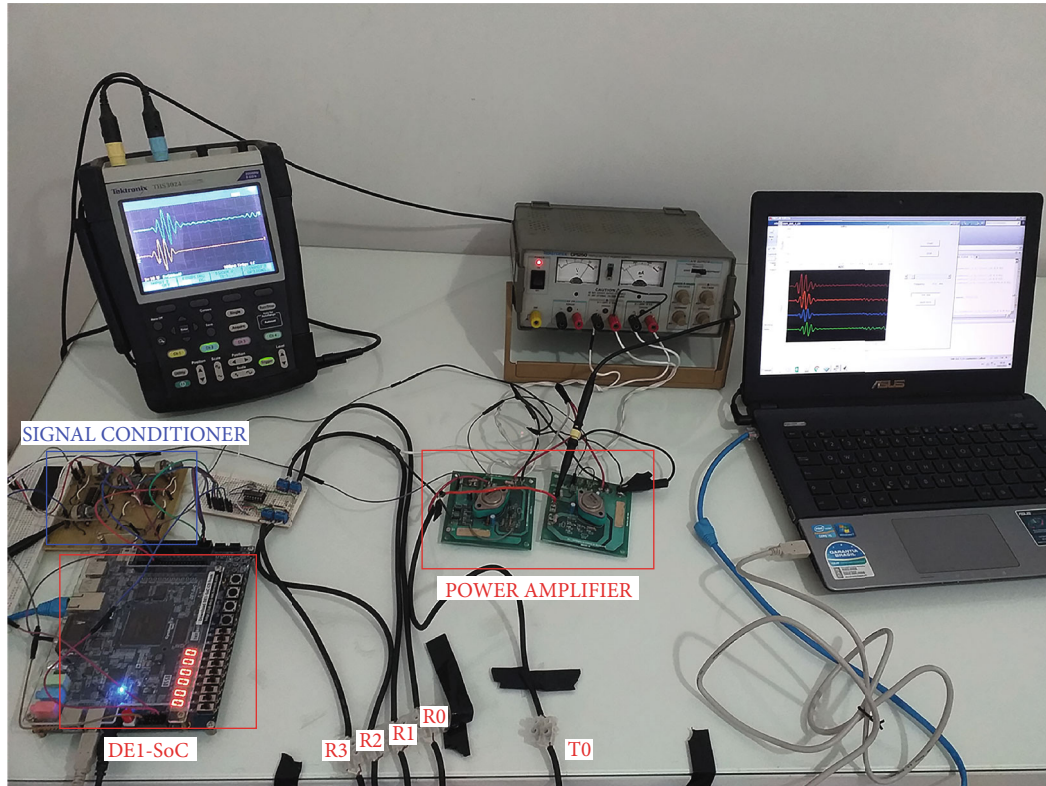
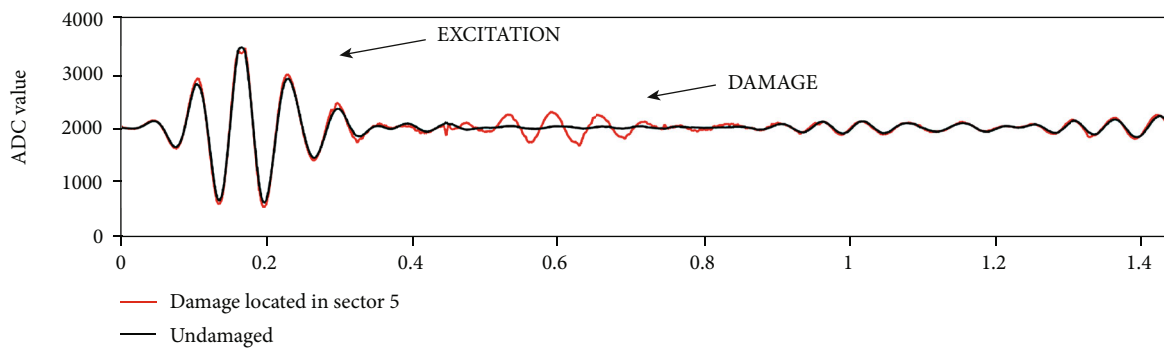
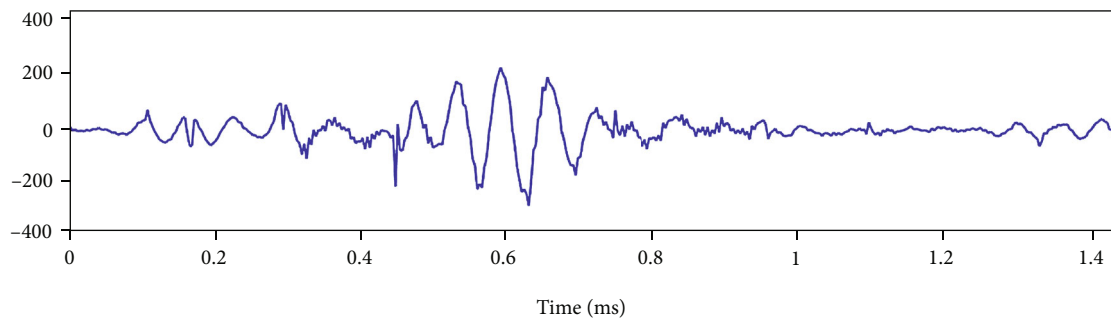


FIGURE 4: Experiment setup.



(a)



(b)

FIGURE 5: Signal processing: (a) Lamb wave signals received by transducer R0; (b) difference signal.

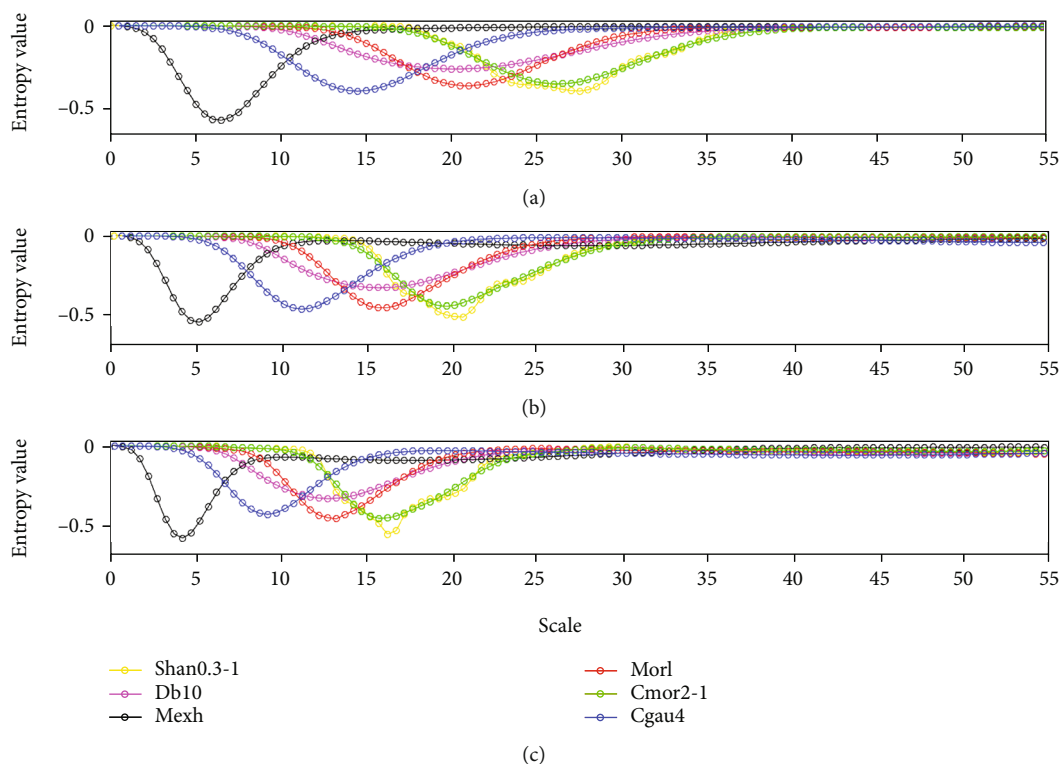


FIGURE 6: CWT entropy: (a) entropy ($F = 12$ kHz); (b) entropy ($F = 16$ kHz); (c) entropy ($F = 20$ kHz).

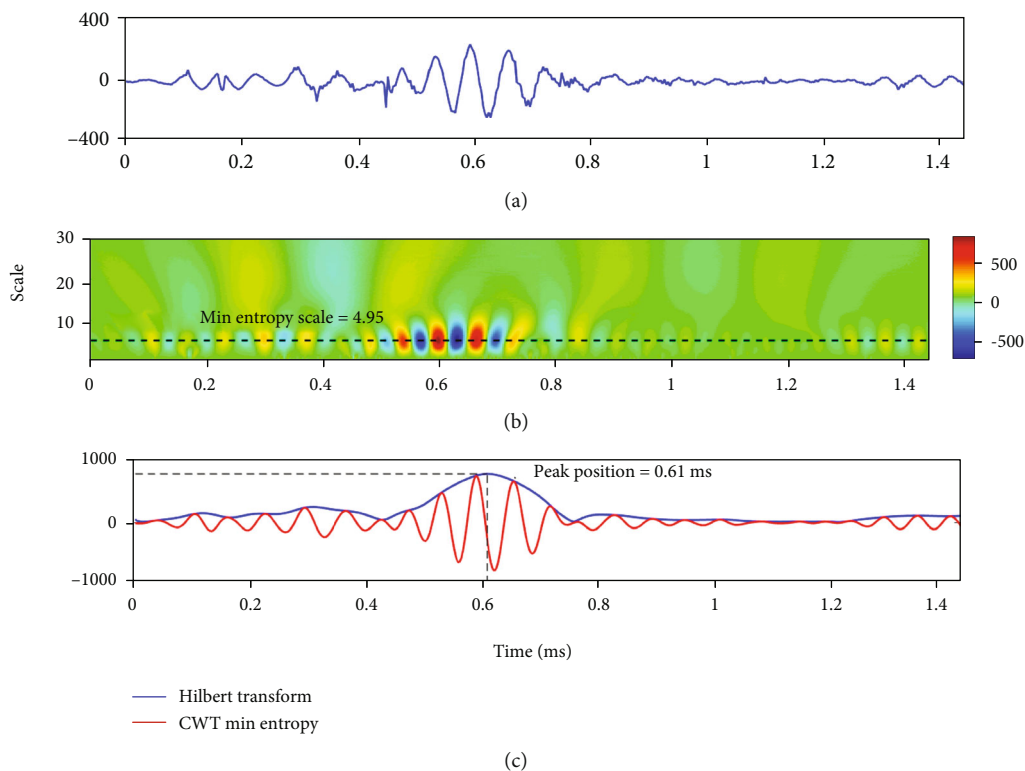


FIGURE 7: CWT signal processing: (a) difference signal; (b) CWT (continuous wavelet transform); (c) CWT minimum entropy.

TABLE 1: ANN input parameters.

ANN inputs	Sensor	Frequency
Input 01	R0	12 kHz
Input 02	R1	12 kHz
Input 03	R2	12 kHz
Input 04	R3	12 kHz
Input 05	R0	16 kHz
Input 06	R1	16 kHz
Input 07	R2	16 kHz
Input 08	R3	16 kHz
Input 09	R0	20 kHz
Input 10	R1	20 kHz
Input 11	R2	20 kHz
Input 12	R3	20 kHz

could be max-pooling or average pooling. The fully connected layer corresponds to MLP-NN.

2.5. Optimal Neural Network Architecture by Multiple Particle Collision Algorithm. The identification of the best ANN configuration for a given application and data is a challenging task. One standard procedure is to do several experiments with different ANN configurations for getting acceptable results. The specialist changes the values of ANN parameters for each trial and compares the results with the observed values until finding a set of the most suitable parameters in his judgment [49]. An alternative approach is to formulate the task of identifying the best ANN topology as an optimization problem [56].

The goal of optimization problems is finding the suitable hyperparameters of a set that maximizes or minimizes a function, defined as an objective function or cost function. The objective function used in this work is given by [57]

$$f_{\text{obj}} = \text{penalty} \times \left(\frac{\rho_1 \times E_{\text{train}} + \rho_2 \times E_{\text{gen}}}{\rho_1 + \rho_2} \right), \quad (14)$$

$$\text{penalty} = C_1 \times e^{(\text{neurons}^2)} + C_2 \times (\text{epochs}) + 1. \quad (15)$$

The first term of Equation (14) is the penalty factor, and the second term is the weighted mean of two errors: train error E_{train} and generalization error E_{gen} . Equation (15) shows the penalty term to evaluate the number of neurons and the epochs to perform the training [49]. The parameter values used in this work are $\rho_1 = 1$, $\rho_2 = 0.1$, $C_1 = 5 \times 10^{-8}$, and $C_2 = 5 \times 10^{-5}$.

The MPCA optimization strategy must find the ANN configuration that minimizes the objective function (Equation (14)).

The ANN parameters taken into account by MPCA are the number of hidden layers, the number of neurons in each hidden layer, the weight values, the learning rate η , the momentum constant α , and the activation function [49].

The MPCA is a version of the particle collision algorithm (PCA). The PCA is a stochastic optimization algorithm

TABLE 2: ANN architecture—MPCA.

Parameter	Value
Hidden layer	1
Hidden layer neuron	5
Learning rate η	0.4
Momentum α	0.3
Activation function	Tanh
Mean squared error	0.002

TABLE 3: 1D-CNN-CWT architecture.

CNN layers
Conv1D(filters = 10, kernel = 3, activation = ReLU)
AveragePooling1D()
Flatten()
Dense(60, activation = ReLU)
Dense(60, activation = Tanh)
Dense(2, activation = ReLU)

TABLE 4: 1D-CNN-RAW architecture.

CNN layers
Conv1D(filters = 10, kernel = 90, activation = ReLU)
Conv1D(filters = 5, kernel = 10, activation = ReLU)
MaxPooling1D()
Flatten()
Dense(60, activation = Tanh)
Dense(2, activation = ReLU)

inspired by the neutron travelling in a nuclear reactor, where two main phenomena can occur: scattering and absorption. The MPCA is a modified version of the particle collision algorithm that uses several particles to explore the search space [48]. Algorithm 1 shows the MPCA pseudocode.

The MPCA starts with a random ANN configuration (Old-Config). The Fitness() function calculates Equation (14) that evaluates the ANN configuration performance and updates the Best-Fitness information. For each particle, a stochastic perturbation (function Perturbation()) generates a new ANN configuration (New-Config) closer than the previous one, and the Fitness() function calculates the fitness of it. If New-Config is better than Old-Config, this new configuration is absorbed (the New-Config becomes an Old-Config for the next iterations). The function Exploration() generates small perturbations on closer positions. However, if New-Config is worse than Old-Config, the function Scattering() sends the particle to a different location of the search space trying to escape the local minimum [56]. The blackboard

TABLE 5: Damage experiments for training.

Sector	(x, y) (mm)	RMSE MPCA-ANN	RMSE 1D-CNN-CWT	RMSE 1D-CNN-RAW
S1	(175, 125)	1.1 mm	1.7 mm	1.9 mm
S2	(125, 125)	2.2 mm	1.6 mm	2.4 mm
S3	(75, 125)	2.4 mm	2.4 mm	3.2 mm
S4	(25, 125)	1.5 mm	2.5 mm	1.5 mm
S5	(175, 75)	1.4 mm	1.4 mm	1.9 mm
S6	(125, 75)	1.8 mm	1.5 mm	2.3 mm
S7	(75, 75)	2.2 mm	1.8 mm	2.0 mm
S8	(25, 75)	2.4 mm	1.3 mm	1.1 mm
S9	(175, 25)	2.3 mm	2.2 mm	1.2 mm
S10	(125, 25)	0.6 mm	1.7 mm	0.9 mm
S11	(75, 25)	1.4 mm	1.8 mm	0.9 mm
S12	(25, 25)	0.7 mm	2.3 mm	1.1 mm

strategy provides the coordination between the multiple particles sharing the Best-Fitness information.

3. Experimental Setup and Results

The validation scenario of the proposed damage detection method is an aluminum plate (500 mm \times 500 mm \times 1 mm) with four PZT transducers bounded on its surface, as represented by Figure 3. These transducers, which have a 20 mm diameter, are arranged with a space of 5 mm and labeled as R0, R1, R2, and T0. Figure 3 shows twelve square sectors (50 mm sides) that delimit the damage scan region. The Lamb wave excitation signal is produced by transducer T0 and received by the other four transducers (R0, R1, R2, and R3).

Figure 4 shows the experiment setup. The DE1-SoC (System-on-Chip) Development board has a hardware design platform which combines the dual-core Cortex-A9 embedded cores with FPGA (Field Programmable Gate Array). This board is programmed to generate a Gaussian window five cycles sine tone burst through a look-up table implemented in programmable logic and using a DAC (10 bits) converter. The DAC signal is amplified by the power circuit; then, the output signal generates the Lamb waves through the PZT transducer T0. The sine signal is generated with three different frequencies: 12 kHz, 16 kHz, and 20 kHz. The Lamb waves propagate through the aluminum board, and they are captured by the PZT transducers R0, R1, R2, and R3. The 12-bit ADC converts the analog voltage from the PZT transducers into digital signals. The sampling rate is set to 400 kHz.

Two signals received by sensor R0 are plotted in Figure 5(a). The first signal (black color) corresponds to the undamaged condition, and the second signal (red color) represents the signal received when damage occurs on sector S5. The excitation arrow indicates the direct wave received from the T0 transducer, and the damage arrow indicates the reflected wave by the damage, as shown in Figure 5(a). Figure 5(b) shows the difference signal, corresponding to the damage signal subtracted from the undamaged signal.

In the next step, the continuous wavelet transform of the difference signal is performed using six different mother wavelets. Figure 6 presents the normalized Shannon entropy of the CWT coefficients. The selected mother wavelet is the mexh (Mexican hat) which has the minimum entropy.

The resultant scalogram is plotted in Figure 7(b). The CWT minimum entropy signal, shown in Figure 7(c), is constructed from the scalogram by getting the CWT coefficients which scale has the minimum entropy value. The Hilbert transform of the CWT minimum entropy signal makes it possible to obtain the peak position as presented in Figure 7(c).

Table 1 shows the ANN inputs. The peak position of the minimum entropy CWT coefficient obtained for each sensor R0, R1, R2, and R3 is the ANN inputs. The ANN output is the damage coordinate (x, y).

A mass of lead is employed to simulate damage in the aluminum plate. The training and validation databases are generated by placing the mass of lead in the center of each sector defined in Figure 3. A total of 3600 samples, 300 samples by sector, is used to train the networks. The total number of samples is randomly divided into three sets: 2520 samples are selected for training, 540 samples for validation, and 540 samples for generalization. The training set is used to update the ANN weights and biases. The validation set evaluates the ANN generalization performance during the training phase (cross-validation process). A poor generalization performance is signaled by the raising of the validation error [56]. The generalization set is used to evaluate the ANN generalization performance after the training process.

Table 2 presents the ANN parameters found by the MPCA optimization algorithm. The parameters used in the MPCA are 10 particles using a multiprocessing machine (one particle per processor) and 30 iterations (a scheme used to explore a better solution around the new particle location), and the stopping criterion is the maximum of 300 objective function evaluations.

The MPCA-ANN results are compared with two 1D-CNN. The 1D-CNN-CWT uses the same dataset applied to MPCA-ANN. This dataset corresponds to the features extracted using the CWT as defined in Table 1. The 1D-CNN-RAW uses the raw signals acquired by the PZT sensors. The mean square

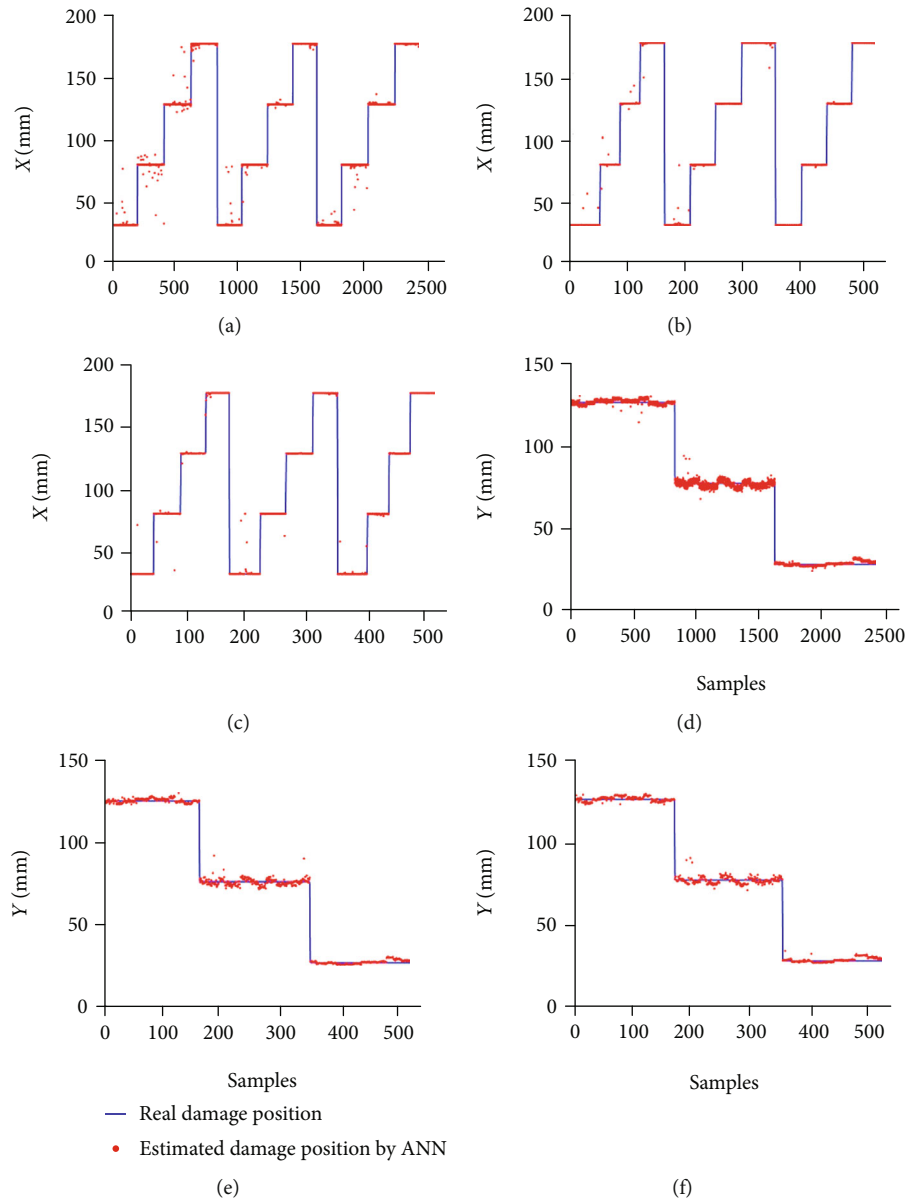


FIGURE 8: MPCA-ANN training performance: (a) training data (X coordinate); (b) validation data (X coordinate); (c) generalization data (X coordinate); (d) training data (Y coordinate); (e) validation data (Y coordinate); (f) generalization data (Y coordinate).

TABLE 6: Damage experiments to verify the generalization.

Damage	(x, y) (mm)	RMSE MPCA-ANN	RMSE 1D-CNN-CWT	RMSE 1D-CNN-RAW
V01	(50, 100)	17.2 mm	11.7 mm	10.8 mm
V02	(100, 100)	14.4 mm	12.8 mm	8.3 mm
V03	(150, 100)	10.3 mm	10.5 mm	4.8 mm
V04	(50, 50)	17.4 mm	9.3 mm	12.1 mm
V05	(100, 50)	10.2 mm	3.3 mm	8.3 mm
V06	(150, 50)	6.6 mm	8.5 mm	11.0 mm

error is used as a loss function with the Adam optimization algorithm. The 1D-CNN design is developed using TensorFlow and Keras environment. TensorFlow is a platform developed

by Google and dedicated to deep learning applications. Keras is a Python high-level deep learning library that runs on top of TensorFlow [8].

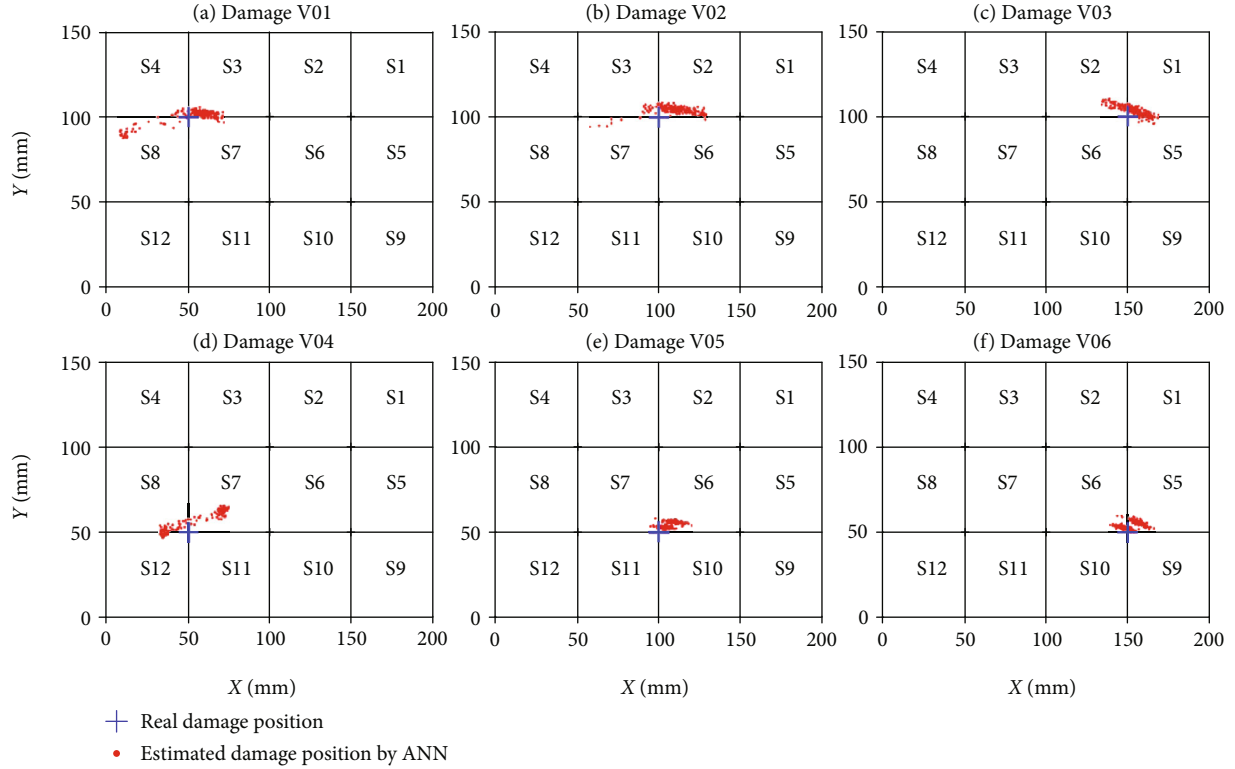


FIGURE 9: MPCA-ANN generalization performance: (a) damage V01; (b) damage V02; (c) damage V03; (d) damage V04; (e) damage V05; (f) damage V06.

A batch size of 32, learning rate of $1e-4$, and 1200 epochs are the training parameters of 1D-CNN-CWT. Table 3 shows the 1D-CNN-CWT architecture.

For 1D-CNN-RAW, the training parameters are batch size of 32, learning rate of $1e-4$, and 200 epochs. Table 4 shows the 1D-CNN-RAW architecture.

Table 5 shows the dataset division by sector. The damage is simulated placing the mass in the center of the sector indicated by the first column of Table 5. The second column of Table 5 presents the damage location represented by Cartesian's coordinates which origin is the corner of sector S12. The last three columns of Table 5 present the root-mean-square error (RMSE) of the estimated damage position by MPCA-ANN, 1D-CNN-CWT, and 1D-CNN-RAW, respectively. Equation (16) calculates the RMSE of the damage positions, where $(x_{\text{ann}}(n), y_{\text{ann}}(n))$ is the damage position estimated by ANN and $(x(n), y(n))$ is the real damage position:

$$\text{RMSE} = \sqrt{\frac{1}{2 \cdot N} \sum_{n=1}^N (x_{\text{ann}}(n) - x(n))^2 + (y_{\text{ann}}(n) - y(n))^2}. \quad (16)$$

Figure 8 shows the MPCA-ANN performance. The real damage position (the mass position in the aluminum board) is represented by blue color, and the estimated damage position by ANN is represented by red color.

In order to evaluate the MPCA-ANN generalization capability, experiments were performed with mass placed in the vertices of sectors S6 and S7. The second column of Table 6 presents each vertex location represented by Cartesian's coordinates which origin is the corner of sector S12. 200 experiments were performed for each vertex. The last three columns of Table 6 present the RMSE of the estimated damage position by MPCA-ANN, 1D-CNN-CWT, and 1D-CNN-RAW, respectively.

Figure 9 shows the damage position estimated by MPCA-ANN for each case of Table 6. Each plot of Figure 9 represents the physical aluminum board; the red points show the estimated damage position and the blue cross the real damage position.

4. Discussion

The minimum Shannon entropy selected the Mexican hat as the best choice for the mother wavelet as shown by Figure 6. The CWT performance can be visualized by Figures 7(b) and 7(c). The results presented by these figures indicate that the CWT can be evaluated only on the scale with minimum entropy value, where more energy concentration exists. The comparison between Figures 7(a) and 7(b) shows that the CWT using the Mexican hat as the mother wavelet, calculated in the scale with minimum entropy, can estimate the peak position of the Lamb wave reflection signal.

The MPCA algorithm found the ANN architecture shown in Table 2. The ANN training performance result, presented in Figure 8, indicates that the ANN can find the correct damage

TABLE 7: Mean normalized distance error.

Algorithm	MND
MPCA-ANN	0.124
D-CNN-CWT	0.082
D-CNN-RAW	0.081

TABLE 8: Estimated computational cost comparison.

	RMSE MPCA-ANN	RMSE 1D- CNN-CWT	RMSE 1D- CNN-RAW
Number of convolutions			
CWT convolution	12	12	—
Convolutional layer 1	—	10	120
Convolutional layer 2	—	—	50
Total	12	22	170
Number of neurons			
Hidden layer 1	5	60	60
Hidden layer 2	—	60	—
Output layer	2	2	2
Total	7	122	62
Number of weights			
Convolutional layer 1	—	40	10810
Convolutional layer 2	—	—	505
Hidden layer 1	65	3060	7560
Hidden layer 2	—	3660	—
Output layer	12	122	122
Total	77	6882	18977

position. Table 5 shows the training results for the MPCA-ANN, 1D-CNN-CWT, and 1D-CNN-RAW. The ANN-MPCA presented a good performance in comparison with the two versions of 1D-CNN. The results of Table 6 show the good generalization capability of MPCA-ANN when compared with the 1D-CNN-CWT and 1D-CNN-RAW.

A metric termed mean normalized distance (MND) [13] is used to compare the global result of the generalization performance. The distance values are normalized with length of the inspected board region. Equation (17) describes the MND metric:

$$\text{MND} = \frac{\sum_{n=1}^N \sqrt{(x_{\text{ann}}^n(n) - x^n(n))^2 + (y_{\text{ann}}^n(n) - y^n(n))^2}}{N}, \quad (17)$$

where $(x_{\text{ann}}^n, y_{\text{ann}}^n)$ is the normalized damage position found by ANN, (x^n, y^n) is the real normalized position of damage, and N is the number of samples. The normalized coordinate x^n corresponds to the division of x by 200 mm (length of inspected area), and the normalized coordinate y^n corresponds to the division of y by 150 mm (length of inspected area). Table 7 shows MND values for the three algorithms, with errors of order $O(10^{-1})$, and the best performance was the 1D-CNN-RAW.

The MND results presented in Table 7 show that 1D-CNN-RAW with features extracted directly from raw data presents the same generalization performance in comparison with the 1D-CNN-CWT. This result shows the good damage localization information contained in ToF features extracted using the CWT optimized by entropy calculation. Normally, the CWT performs several convolutional operations using different scale values. The CWT operation has high computational cost. However, the entropy method limits the CWT just to optimum scale value reducing the computational cost of the CWT operation. Therefore, the 1D-CNN-CWT has the same performance as 1D-CNN-RAW but with a low computational cost. The 1D-CNN-RAW uses two convolutional layers, while 1D-CNN-CWT uses just one convolutional layer. Table 8 shows an estimated computational cost comparison between MPCA-ANN, 1D-CNN-CWT, and 1D-CNN-RAW. The computational cost is indirectly estimated by the number of convolutions, the number of neurons, and the number of weights.

Both 1D-CNN methods presented a better generalization performance in comparison with MPCA-ANN. This result confirms the great 1D-CNN capability to extract relevant features from input data. However, the 1D-CNN structure has a high computational cost. The 1D-CNN-CWT architecture, presented in Table 3, has two fully connected layers with 60 neurons, while the 1D-CNN-RAW architecture, presented in Table 4, has one fully connected layer with 60 neurons. The generalization performance of MPCA-ANN with only one hidden layer and five neurons was close to the complex 1D-CNN structures. An estimated computational cost is presented in Table 8. The lowest computational effort is linked to the MPCA-ANN, due to the design based on searching the simplest neural architecture. Such low neural network complexity shows the adequacy for embedded applications, with limited computational capacity, or to be implemented in a hardware processing device, maintaining a good generalization capability.

5. Conclusion

In this paper, an optimized artificial neural network is proposed to locate damages into plate-like structures. The Multiple Particle Collision Algorithm finds the most adequate ANN architecture for the considered dataset. The experimental results of damage detection and localization in an aluminum plate validate the effectiveness of MPCA in finding a simple and optimized ANN architecture with good generalization capability. The generalization performance of MPCA-ANN was compared with two 1D-CNN. The results

show that MPCA-ANN has a good performance in comparison with complex 1D-CNN structures.

The continuous wavelet transform processes the Lamb wave reflections and improves the time-of-flight estimation. The Shannon entropy-based criterion finds the best mother wavelet and scale values for detecting the Lamb wave reflections. The experimental results validate the effectiveness of the proposed optimized CWT. In future works, this method will be extended to more complex plate-like structures such as composite structures.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, Finance Code 001.

References

- [1] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure prognosis and applications - a survey of recent literature," *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 728–748, 2019.
- [2] J. Zhang, H. Sun, Z. Sun, W. Dong, and Y. Dong, "Fault diagnosis of wind turbine power converter considering wavelet transform, feature analysis, judgment and BP neural network," *IEEE Access*, vol. 7, pp. 179799–179809, 2019.
- [3] M. Mousavi, M. Moradi, A. Chaibakhsh, M. Kordestani, and M. Saif, "Ensemble-based fault detection and isolation of an industrial gas turbine," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 2351–2358, Toronto, ON, Canada, 2020.
- [4] N. Mukherjee, A. Chattopadhyaya, S. Chattopadhyay, and S. Sengupta, "Discrete-wavelet-transform and Stockwell-transform-based statistical parameters estimation for fault analysis in grid-connected wind power system," *IEEE Systems Journal*, vol. 14, no. 3, pp. 4320–4328, 2020.
- [5] V. Veerasamy, N. I. A. Wahab, M. L. Othman et al., "LSTM recurrent neural network classifier for high impedance fault detection in solar PV integrated power system," *IEEE Access*, vol. 9, pp. 32672–32687, 2021.
- [6] B. Rathore, O. P. Mahela, B. Khan, H. H. Alhelou, and P. Siano, "Wavelet-alienation-neural-based protection scheme for STATCOM compensated transmission line," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2557–2565, 2021.
- [7] Z. S. Khodaei and M. H. Aliabadi, "A multi-level decision fusion strategy for condition based maintenance of composite structures," *Materials*, vol. 9, no. 9, p. 790, 2016.
- [8] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, "A review of vibration-based damage detection in civil structures: from traditional methods to machine learning and deep learning applications," *Mechanical Systems and Signal Processing*, vol. 147, article 107077, 2021.
- [9] S. Sundararaman, D. E. Adams, and E. J. Rigas, "Structural damage identification in homogeneous and heterogeneous structures using beamforming," *Structural Health Monitoring*, vol. 4, no. 2, pp. 171–190, 2005.
- [10] A. Rai and M. Mitra, "Lamb wave based damage detection in metallic plates using multi-headed 1-dimensional convolutional neural network," *Smart Materials and Structures*, vol. 30, article 035010, 2021.
- [11] L. Huang, L. Zeng, J. Lin, and N. Zhang, "Baseline-free damage detection in composite plates using edge-reflected Lamb waves," *Composite Structures*, vol. 247, article 112423, 2020.
- [12] S. W. F. de Rezende, J. D. R. V. de Moura, R. M. F. Neto, C. A. Gallo, and V. Steffen, "Convolutional neural network and impedance-based SHM applied to damage detection," *Engineering Research Express*, vol. 2, no. 3, 2020.
- [13] S. Zhang, C. M. Li, and W. Ye, "Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network," *Mechanical Systems and Signal Processing*, vol. 147, article 107107, 2021.
- [14] E. Dragašius, D. Eidukynas, V. Jūrėnas et al., "Piezoelectric transducer-based diagnostic system for composite structure health monitoring," *Sensors*, vol. 21, p. 253, 2021.
- [15] S. Yuan, H. Wang, and J. Chen, "A PZT based on-line updated guided wave - Gaussian process method for crack evaluation," *IEEE Sensors Journal*, vol. 20, no. 15, pp. 8204–8212, 2020.
- [16] M. S. Hameed, Z. Li, J. Chen, and J. Qi, "Lamb-wave-based multistage damage detection method using an active PZT sensor network for large structures," *Sensors*, vol. 19, no. 9, p. 2010, 2019.
- [17] M. Rautela and S. Gopalakrishnan, "Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks," *Expert Systems With Applications*, vol. 167, 2021.
- [18] V. Giurgiutiu, *Structural Health Monitoring with Piezoelectric Wafer Active Sensors*, Academic Press, 2nd ed. edition, 2008.
- [19] P. Kudela, M. Radzienski, W. Ostachowicz, and Z. Yang, "Structural health monitoring system based on a concept of Lamb wave focusing by the piezoelectric array," *Mechanical Systems and Signal Processing*, vol. 108, pp. 21–32, 2018.
- [20] M. Azimi, A. D. Eslamlou, and G. Pekcan, "Data-driven structural health monitoring and damage detection through deep learning: state-of-the-art review," *Sensors*, vol. 20, no. 10, p. 2778, 2020.
- [21] F. Li, H. Peng, and G. Meng, "Quantitative damage image construction in plate structures using a circular PZT array and lamb waves," *Sensors and Actuators, A: Physical*, vol. 214, pp. 66–73, 2014.
- [22] O. Saotome, S. V. Farias, E. G. O. Nóbrega, I. Sander, and D. S. Loubach, "Structure health monitoring using discrete wavelet transform and piezoelectric sensor array," in *Aerospace Technology Congress*, pp. 1–7, Solna, Stockholm, 2016.
- [23] X. Qing, W. Li, Y. Wang, and H. Sun, "Piezoelectric transducer-based structural health monitoring for aircraft applications," *Sensors*, vol. 19, no. 3, p. 545, 2019.
- [24] S. V. Farias, O. Saotome, E. G. O. Nóbrega, I. Sander, and D. S. Loubach, "Multiple damage detection in plate structure using piezoelectric array sensors and 2D-MUSIC spectrum," in *Aerospace Technology Congress*, pp. 1–6, Solna, Stockholm, 2016.

- [25] Q. Bao, S. Yuan, and F. Guo, "A new synthesis aperture-MUSIC algorithm for damage diagnosis on complex aircraft structures," *Mechanical Systems and Signal Processing*, vol. 136, article 106491, 2020.
- [26] T. Fu, Y. Wang, L. Qiu, and X. Tian, "Sector piezoelectric sensor Array transmitter beamforming MUSIC algorithm based structure damage imaging method," *Sensors*, vol. 20, no. 5, p. 1265, 2020.
- [27] M. Mitra and S. Gopalakrishnan, "Guided wave based structural health monitoring: a review," *Smart Materials and Structures*, vol. 25, no. 5, article 053001, 2016.
- [28] N. Mori, S. Biwa, and T. Kusaka, "Damage localization method for plates based on the time reversal of the mode-converted Lamb waves," *Ultrasonics*, vol. 91, pp. 19–29, 2019.
- [29] Z. Li, "Transverse damage localization and quantitative size estimation for composite laminates based on Lamb waves," *IEEE Access*, vol. 7, pp. 174859–174872, 2019.
- [30] B. Feng, D. J. Pasadas, A. L. Ribeiro, and H. G. Ramos, "Locating defects in anisotropic CFRP plates using ToF-based probability matrix and neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 5, pp. 1252–1260, 2019.
- [31] L. D. Chiwiacowsky, E. H. Shiguemori, H. F. Campos Velho, P. Gasbarri, and J. D. Silva, "A comparison of two different approaches for the damage identification problem," *Journal of Physics: Conference Series*, vol. 124, no. 1, 2008.
- [32] Z. Zhang, H. Pan, X. Wang, and Z. Lin, "Machine learning-enriched lamb wave approaches for automated damage detection," *Sensors*, vol. 20, no. 6, p. 1790, 2020.
- [33] J. B. Harley and D. Sparkman, "Machine learning and NDE: past, present, and future," in *AIP Conference Proceedings*, vol. 2102, pp. 090001–1–090001–10, Vermont, USA, 2019.
- [34] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: a survey," *Mechanical Systems and Signal Processing*, vol. 151, article 107398, 2021.
- [35] A. Gülcü and Z. Kuş, "Multi-objective simulated annealing for hyper-parameter optimization in convolutional neural networks," *PeerJ Computer Science*, vol. 7, p. e338, 2021.
- [36] Y. Lu, L. Ye, Z. Su, L. Zhou, and L. Cheng, "Artificial neural network (ANN)-based crack identification in aluminum plates with lamb wave signals," *Journal of Intelligent Material Systems and Structures*, vol. 20, no. 1, pp. 39–49, 2009.
- [37] N. P. Yelve and I. B. Mulla, "Locating damage in thin metallic plates using lamb waves and artificial neural network," in *2017 International Conference on Nascent Technologies in Engineering, ICNTE 2017- Proceedings*, pp. 1–7, Vashi, India, 2017.
- [38] D. F. Hesser, G. K. Kocur, and B. Markert, "Active source localization in wave guides based on machine learning," *Ultrasonics*, vol. 106, article 106144, 2020.
- [39] J. A. Anochi, R. Hernández Torres, and H. F. de Campos Velho, "Two geoscience applications by optimal neural network architecture," *Pure and Applied Geophysics*, vol. 177, no. 6, pp. 2663–2683, 2019.
- [40] B. Jamali, M. Rasekh, F. Jamadi, R. Gandomkar, and F. Makiabadi, "Using PSO-GA algorithm for training artificial neural network to forecast solar space heating system parameters," *Applied Thermal Engineering*, vol. 147, pp. 647–660, 2019.
- [41] B. Miao, M. Wang, S. Yang, Y. Luo, and C. Yang, "An optimized damage identification method of beam using wavelet and neural network," *Engineering*, vol. 12, no. 10, pp. 748–765, 2020.
- [42] X. Zhang, H. Nguyen, X. N. Bui et al., "Evaluating and predicting the stability of roadways in tunnelling and underground space using artificial neural network-based particle swarm optimization," *Tunnelling and Underground Space Technology*, vol. 103, article 103517, 2020.
- [43] A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, "An efficient hybrid multilayer perceptron neural network with grasshopper optimization," *Soft Computing*, vol. 23, no. 17, pp. 7941–7958, 2019.
- [44] P. G. Benardos and G. Á. Vosniakos, "Optimizing feedforward artificial neural network architecture," *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 365–382, 2007.
- [45] S. Soares, C. Henggeler, and R. Araújo, "Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development," *Neurocomputing*, vol. 121, pp. 498–511, 2013.
- [46] S. Kaoutar, "Multi-criteria optimization of neural networks using multi-objective genetic algorithm," in *Intelligent Systems and Computer Vision*, pp. 1–4, Fez, Morocco, 2017.
- [47] V. K. Ojha, A. Abraham, and V. Snásel, "Ensemble of heterogeneous flexible neural trees using multiobjective genetic programming," *Applied Soft Computing*, vol. 52, pp. 909–924, 2016.
- [48] F. P. da Luz, "A new multi-particle collision algorithm for optimization in a high performance environment," *Journal of Computational Interdisciplinary Sciences*, vol. 1, pp. 3–10, 2008.
- [49] J. A. Anochi, H. F. Campos Velho, H. C. M. Furtado, and E. F. Luz, "Self-configuring two types of neural networks by MPCA," *Journal of Mechanics Engineering and Automation*, vol. 5, no. 2, pp. 112–120, 2015.
- [50] F. Li, G. Meng, K. Kageyama, Z. Su, and L. Ye, "Optimal mother wavelet selection for Lamb wave analyses," *Journal of Intelligent Material Systems and Structures*, vol. 20, no. 10, pp. 1147–1161, 2009.
- [51] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 2nd ed. edition, 1999.
- [52] P. R. De Souza and E. G. De Oliveira Nobrega, "A fault location method using lamb waves and discrete wavelet transform," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 34, no. 4, pp. 515–524, 2012.
- [53] J. C. Goswami and A. K. Chan, *Fundamentals of Wavelets: Theory, Algorithms, and Applications*, Wiley, 2nd ed. edition, 2011.
- [54] R. N. Bracewell, *The Fourier Transform And Its Applications*, McGraw-Hill, 3rd ed. edition, 2000.
- [55] S. Haykin, *Neural Networks - A Comprehensive Foundation*, Prentice Hall, 2nd ed. edition, 1998.
- [56] J. Anochi and H. F. Campos Velho, "Optimization of feedforward neural network by multiple particle collision algorithm," in *IEEE Symposium on Foundations of Computational Intelligence, FOCI*, pp. 1–4, Orlando, FL, USA, 2014.
- [57] A. R. Carvalho, F. M. Ramos, and A. A. Chaves, "Metaheuristics for the feedforward artificial neural network (ANN) architecture optimization problem," *Neural Computing and Applications*, vol. 20, no. 8, pp. 1273–1284, 2011.