COBEM 2021 26th International Congress of Mechanical Engineering
November 22-26, 2021, Virtual Congress, Brazil

COB-2021-0519
APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO DAMAGE DETECTION OF COMPOSITE MATERIALS

Matheus Janczkowski Fogaça
Eduardo Lenz Cardoso
Ricardo de Medeiros
Santa Catarina State University, Department of Mechanical Engineering, Rua Paulo Malschitzki, 200. Zona Industrial Norte, 89219-710 Joinville, SC, Brazil.
matheusj2009@hotmail.com, eduardo.cardoso@udesc.br, ricardo.medeiros@udesc.br

Abstract. *Composite materials are a recurrent choice in Engineering, since their strength to weight ratio is steadily increasing and there is the possibility to tailor the material properties during construction. Therefore, it is fundamental to develop methodologies to detect damage in these materials. A powerful tool for this task is the vibration analysis, whose outputs may be interpreted with Artificial Neural Network (ANN) to identify and classify whether a sample is damaged or not. This work aims to evaluate and to investigate the relationship between the performance and the parameters of a feed-forward ANN to be applied in damage detection of composite structures. To this end, topology, activation functions, data compressing methods, optimizers, testing, training, and validation data sets are analyzed. Vibration data from Glass Fiber Reinforced Plastic (GFRP) composite beam are used. However, it is not feasible to directly use large dimensional data, i.e., Frequency Response Function (FRF), which could demand more complex models and high computational cost. Therefore, different techniques for dimension reduction, like Principal Component Analysis (PCA), Dislocated Series (DS), and Linear Discriminant Analysis (LDA), are investigated. Julia language, especially the Flux library, is employed to build the ANN, validate, train, and test the models, which classify the samples as intact or damaged. Different damage patterns are generated to evaluate the performance of the methodology. Thus, every aspect of the machine learning process is compared to find which combination of parameters is the best for this application. The preliminary results show that the ANN allied with the vibration analysis in the frequency domain can be a powerful tool to detect damage in Engineering situations before catastrophic failure and without the necessity of expensive assays.*

Keywords: *Machine Learning, Artificial Neural Network, Composite Materials, Data set Analysis, Optimization.*

1. INTRODUCTION

The continual challenge for the industry is to make parts, assemblies, and machinery lighter and more efficient. Thus, there is a progressively wider application of composites due to their excellent strength, stiffness, and corrosion resistance properties (Babu *et al.*, 2016). According to Karsh *et al.* (2018), the failure mechanism of laminated composites is complex and governed by several effects, which can make failures hard to predict. The presence of delamination may severely degrade the stiffness and the strength of composites and, in some cases, can lead to catastrophic failure. Thus, if composite materials are to play a bigger role in the industry, then a reliable integrity assessment system has to be developed (Valdes and Soutis, 1999). The principle behind damage detection methods based on dynamic analysis is that damage changes vibration characteristics of the structures, including Frequency Response Functions (FRF), natural frequencies, and mode shapes, which results in the changes of the resonant frequencies (Adams *et al.*, 1975; Campbell, 2010; Bandara *et al.*, 2014). Although there are several techniques to detect damage, some are slow or even require physical access to difficult places (Farrar and Worden, 2012). Besides that, another problem is the high preventive and corrective cost of maintenance, the latter generating the greatest losses in economy and time, in addition to exposure to hazardous situations. To reduce such problems, the use of damage detection techniques is being increasingly investigated and optimized (Zou *et al.*, 2000; Fan and Qiao, 2011; Zhang *et al.*, 2013; Garcia and Trendafilova, 2014; Zhang *et al.*, 2016; De Medeiros *et al.*, 2018; Khan *et al.*, 2019; Zheng *et al.*, 2021).

On the other hand, Artificial Neural Networks (ANNs) methodology is an attractive mathematical tool, which can be used to simulate a wide diversity of scientific and engineering problems. ANNs have been extensively used in structural engineering applications in the area of failure prediction (Junior *et al.*, 2020), delamination identification (Gomes *et al.*, 2019a; Oliver *et al.*, 2021), crack detection (Pereira *et al.*, 2020). ANNs are based on interconnected neurons, which processes information in a similar way our brain does. The network produces a function that relates input parameters with output responses (Oliver *et al.*, 2021). ANNs can learn from examples and can be trained to find solutions to the

complex non-linear, multi-dimensional function relationships without any prior assumptions about their nature (Zhang and Friedrich, 2003). There are several works using ANNs to identify damage. Yam *et al.* (2003) presented an integrated method for damage detection of composite structures using their vibration responses, wavelet transform, and artificial neural networks (ANN) identification. The results show that the method can be applied to online structural damage detection and health monitoring for various industrial structures. Fang *et al.* (2005) studied structural damage detection using a backpropagation neural network and frequency-domain functions comparing three different learning rate algorithms. Very high accuracy to predict damage location and severity was reached. Zhang *et al.* (2013) examined three different inverse algorithms for solving the non-linear equations to predict the interface, lengthwise location, and size of delamination: direct of solution using a graphical method, artificial neural network (ANN), and surrogate-based optimization. Zhang *et al.* (2018) studied the inverse algorithms to model-measurement discrepancies, different levels of random artificial noise are added to the numerical frequencies, and sensitivity tests on beams with different delamination sizes and interfaces were carried out. Gomes *et al.* (2019b) discussed the use of optimization algorithms and Artificial Neural Networks (ANN) for structural monitoring in the form of a brief review. Oliver *et al.* (2021) presented a methodology for damage detection and identification on laminated composite plates using artificial neural networks fed with modal data obtained by finite element analysis. To overcome problems like a large number of inputs, regarding the FRFs data, one established method in the literature is the Principal Component Analysis (PCA), providing not only data compression but assisting in feature extraction and conditioning (Bishop *et al.*, 1995).

There are some distinct approaches used to detect structural damage, and it is broadly recognized that the success of a certain methodology may depend on the structure being monitored (Völtz, 2019). Consequently, new methodologies should be verified for different mechanical systems and damage cases (Gul and Catbas, 2009). This work is a continuation of the reference (Völtz *et al.*, 2017, 2018, 2019; Reis *et al.*, 2019, 2021). However, the scientific contribution of this manuscript is related to develop a methodology to determine the best arrangement of hyperparameters of ANNs to discover which one gives the greatest accuracy in all data sets. The data was acquired and treated from the experimental analysis of vibration of composite beams in the free-free boundary condition. The material was layers of fiberglass and epoxy resin as its matrix. Delamination damage was imposed upon some samples, in three different scales, while some samples were left intact. The data collected through an accelerometer was converted into the frequency domain with Fourier Transform and was split into three different and independent sets, training, validation, and testing. Later on, the data was compressed using the PCA technique (Principal Component Analysis) and used to train, validate and test the models created with prescribed topologies. To trace a consistent path of study and comparison among the performances of arrangements of hyperparameter, some of them were fixed, such as activation functions, type of topology, and optimization parameters. Some criteria were developed to measure the quality of the result of each topology and to explain why some of them came out to be better for the classification problem studied in this paper. There are some to cite briefly: probability distribution of model's error during training; distribution of accuracy in the validation data set; and accuracy in the testing data set.

2. DEFINITION OF THE CLASSIFICATION PROBLEM

An interest in determining if a structure or item is intact or damaged is self-evident. Therefore artificial neural networks are developed to classify in which category a sample might be. For this study, some of the samples were deliberately damaged in three different ways, with increasing severity. Then two classifications are possible:

1. If a sample is damaged or not.

2. If a sample is intact or, if it is damaged, what is the severity.

The input for the classifier is the magnitude of the Frequency Response Function (FRF), obtained through vibration and non-destructive experiments.

3. METHODOLOGY

This section aims to present the main steps and concepts needed to implement, verify, and quantify an artifical network performance.

3.1 Experimental procedure

A total of 73 laminate composite beams of fiberglass and epoxy resin were built. The dimensions are: length of 227.75 mm, width of 28.31 mm, and thickness of 2.99 mm. Delamination defect was induced in some of the beams by using a *Teflon* strap between middle layers and positioned on the middle line between the extremities, therefore extending through the whole width of the beam. Defect sizes, thus the width of the Teflon strap, are: 5 mm (damage I or D1), 10 mm (damage II or D2) and 19 mm (damage III or D3). Considering the whole set of beams, 25 were left intact, 15 were of the D1 group, 16 were of the D2 group and 17 of the D3 group.

To avoid errors originated from fixation devices, like the ones in cantilever beams, this experiment was lead with a free-free boundary condition. Each sample was hit with an impact hammer model 8206-003 (sensitivity 1.14 mV/N) and the Frequency Response Function (FRF) and natural frequencies were obtained via an accelerometer model 4517-C. This particular device was placed out of nodal lines, as their locations were already known and estimated through modal analysis. The collected data comprehends an ensemble of 6400 spectral points in the range of 0-3200 Hz. More details about the experimental procedure can be found in Völtz (2019) and Reis (2020).

3.2 Data splitting and compressing procedures

The data must be split to expose the network during training to a fraction of the original set. This is important to appraise if the model can classify the other samples using just the information previously learned. Otherwise there would be contamination in the learning process from other sets and, then, the classification of these sets apart from training would be biased, leading to a false measure of accuracy. The partition of the entire data acquired followed the conventional rule of three independent sets, training, validation, and testing. The percentage of which one takes from the whole was settled in 75%, 15%, and 10% respectively. These percentages are in agreement with the specifications of other works, as Looney (1996), who suggested 65% for training, 10% for validation, and 25% for testing. While (Silva *et al.*, 2010) recommended 60-90% for training and validation. Extracting an average of those ranges, it came about the percentage conducted on this work, that is close to the suggestion presented in another work, 70%, 20%, and 10%, regarding the same order of sets conventioned earlier by Pereira and Bezerra (2007).

As the number of points in the frequency domain is large, the use of the raw data leads to a large number of inputs to the neural network. Thus, it is advisable to reduce the dimensionality of the data, while maintaining the quality of the information. The number of connections would lead to more difficulty in the training process, as the number of weights and biases would be huge. According to in Völtz (2019), the main reason to compress the data or, more specifically, reduce its dimensionality, is to mitigate the worst effects of the curse of dimensionality Bishop *et al.* (1995). Thus the goal is to achieve a way where most variability is preserved whilst the quantity of dimensions is reduced. Principal Component Analysis (PCA) is a technique to achieve this objective, calculating a basis of eigenvectors of the covariance matrix and then making a basis change to a new space with smaller dimensionality.

In this study, 4400 spectral points were selected from the raw set. It was separated 199 points ahead of the first one to cut off the influence of noise present in the lowest frequencies. The same reason was used to exclude the last 1800 points. However, it is important to highlight that all the peaks on FRF were preserved within this interval. Although the dimensionality has already decreased to 4400 by a cut of the whole set, just as if Dislocated Series method were used to create a single mini-batch, it must be diminished further on, then the PCA method comes in. Actually with 30 principal components, thereby 30 eigenvectors sorted by the norm of their eigenvalues, more than 98% of the variance of the data set was kept, which is remarkable as almost all variance is maintained while dimensionality has shrunk 147 times.

The question is wherefrom calculate the covariance matrix at the first point, whether in the whole data set or the slice of the training set. As cited in Völtz (2019), according to Karpathy (2017), Ng (2018), and Vidhya (2016), evaluating the covariance matrix of the entire data set leads to contamination of information from the samples of one subset to another. Consequently, the best approach to preserve one of the three sets to gain information from another is to split the whole first, apply PCA onto the training set and reuse the same matrix of eigenvectors to operate the basis change of the validation and testing sets to the maximized variance space. Thus there are in the end three of them, as expected, with the same number of variables, 30. Using the partition stated previously, the training, validation, and testing sets gained each 55, 11, and 7 samples, respectively. The proportions of intact, D1, D2, and D3 beams were equally scattered along with them.

3.3 Hyperparameters configuration

To perform a comparison among arrangements of hyperparameters, some of them were kept fixed to improve performance, such as 5000 iterations during training's optimization, use of ADAM optimizer with a learning rate of 0.3. Some hyperparameters were varied during the first stage of the research and thus were trained fewer topologies with such configuration, for example, pure relu and relu intercalated with sigmoid activation function arrays were tested, but they revealed themselves as bad options as relu messed with the optimization process and made it too difficult to find a minimum. Therefore, only the sigmoid activation function was analyzed and discussed. Another example is the usage or not of regularization techniques, L1 and L2 have been shown not fit for this particular task, as they induce the same issue of relu. Thus there was employed no regularization method.

The presence of softmax function in the last layer was also studied and it was revealed easier to reduce the loss function without it, however, the least loss values found with softmax in this position were much smaller than those gotten without this function. So the conclusion is softmax gives more accuracy in classification problems in which there are two or more output neurons and the expected output is bounded between 0 and 1, which can be explained by the nature of softmax, for it is a probability distribution giving function. Hence all topologies trained in this study used these configurations.

Loss function was another important topic and as such mean square error (MSE) and cross-entropy are commonly

used and were also tested. MSE has given worse results and has turned out to be less effective in the training, thus cross-entropy function was employed in most ensembles tested in this study. Regarding network architecture, all topologies trained were dense, *i.e.*, all neurons in a layer i connected with all neurons in a layer i-1.

3.4 Topology configuration

A genetic approach was used by Völtz (2019) to find the best performing topology in training and validation data sets, with this technique the model with greater accuracy and generalization ability is selected and used to build up improved versions of itself or mixed with another good model and the crossover of them creates a new topology. This particular method was not fit for this study because there would have been tested many topologies in order to find the best one, and the question regarding why it's the best option is key, consequently, there must be created a way to search a criterion to diagnose the reasons behind a particular performance.

Before determining the array of topologies, it must be set its boundary conditions, *i.e.*, number of neurons on input and output layers. For the input layer, it is equal to the dimensionality of data, 30 for this case, since that is what came out of PCA. The output layer is trickier and has the power to impact on model's performance, according to the first experiments, the most satisfying results come when the output layer has the number of neurons equal to the number of classification cases. As described in section 2., two ways of classifying and addressing the problem are possible, due to the existence of three classes of damage (D1, D2, and D3). For the first way, there must be two neurons on the output layer, because there are two classifications - intact or damaged - on the other hand, for the second option, described in the second enumerated item of section 2, four neurons have to be present on this layer, because there are four classifications, intact, D1, D2 and D3. Two tables follow in order to demonstrate how the output values were set to be, Table 1 exhibits for the first approach, *i.e.*, two neurons in the output layer, while Table 2 displays the second approach. These are restricted between 0 and 1 to match the probabilistic nature of the softmax function.

Table 1. Classification cases and outputs for the first approach of section 2.

Classification	**Output neuron I**	**Output neuron II**
Intact	1	0
Damaged	0	1

Table 2. Classification cases and outputs for the second approach of section 2.

Classification	**Output neuron I**	**Output neuron II**	**Output neuron III**	**Output neuron IV**
Intact	1	0	0	0
D1	0	1	0	0
D2	0	0	1	0
D3	0	0	0	1

Alike the procedure adopted by Reis (2020), inserting a residue in those outputs in training data set, so 1 turns into 0.9999 and 0 into 0.0001, improves the training process and leads to better generalization as it relaxes the desired target on the loss function, since the output is obligatorily restricted in a range between 0 and 1 and it will likely not reach these extreme values.

Since the boundary conditions are discovered, a path has to be tracked to reliably and consistently determine the topologies. For that it was used polynomials in the form of

$$p(x) = a(x - c)^n + b, \tag{1}$$

where *n* is an integer in the interval -1, 0, 1, 2, 3... or the inverse of a natural number, *a* and *b* are constant coefficients and *c* is the position of a critical point in case there is one. If n is equal to 0 or 1, there is no concavity at all. In case it lies on the interval 2, 3, 4..., the polynomial will have a concavity pointing downwards. On the other hand, if n is equal to -1 or is the inverse of a natural number, then the polynomial will present a concavity pointing upwards.

Let *m* be the number of hidden layers that a network is meant to have. If the domain of the polynomial function expressed in Eq. 1 is the position of the layers, let the input one be on $x = 1$, the output one on $x = m + 2$, and the i-th hidden layer on $x = i + 1$. With this settled down, the calculation of coefficients *a* and *b* of Eq. 1 shall be done with the solution of a linear system of equations, formed by the Vandermonde matrix adapted to this particular case and with the vector of prescribed values of $p(x)$, which equals to those numbers of neurons in the input and output layers. At this point, it can be clarified that the goal here is to build a mathematical instrument to plot the topology onto a frame of layers, whose lateral boundaries create a rectangle with height equal to the difference between the number of neurons in the input

and output layers and with a length equal to the number of hidden layers plus one. As it might be seen, a monotonically descending curve restricted inside this rectangle is a feed-forward network. And, in truth, these polynomials give the ability to sweep through the feed-forward case in much fewer topology ensembles than if it would be dealt with the hundreds of possibilities of the number of neurons per layer.

Notice that because a feed-forward network is enclosed within this rectangle, constant c from Eq. 1 has to be null, to maintain a monotonically descending curve, as illustrated on the graph present at Figure 1(a). Thus all feed-forward topologies were trained with c equal to 0. Differently, deep feedforward networks have to have c equal to a number that informs the position of the layer where the maximum is located, like the second layer on Figure 1(a). Another fundamental difference between the two graphs contained in Figure 1 is the technique used to generate the intermediate curves. In truth, the left graph was generated by the usage of Eq. 1 nine times, and the polynomial degree, n, is expressed on the legend with the color of its curve. The contour with $n = 0$ was made to create performance criteria among topologies with a different number of hidden layers, therefore all those layers have the same number of neurons, which is equal to the mean of the numbers of input and output. Then using this architecture, with hidden layers carrying the same characteristic, the choice of whether how many intermediate layers to use can be qualified.

On the other hand, the graph on Figure 1(b) was generated by bisecting two given curves $2^k - 1$ times, where *k* stands for the number of divisions. The function resulted from the bisection of two adjacent contours is

$$p_{j,k}(x) = p_{j-1,k}(x)^f p_{j+1,k}(x)^g, \tag{2}$$

where *j* is the order of a curve from bottom to top in the same direction as of the neurons-axis, f and g are rational exponents whose sum has to be equal to 1. Another possibility for the bisection procedure is with a linear combination,

$$p_{j,k}(x) = vp_{j-1,k}(x) + wp_{j+1,k}(x), \tag{3}$$

in which *v* and *w* are the constants of the linear combination. Alike f and g, v and w must have their sum equal to 1, and can be used to pull the bisected curve nearer to one of the initially given contours. These, following the ordering sequence dictated on both graphs of Figure 1, named as the *inferior limit*, *I.L*, and *superior limit*, *S.L*. The initials are used on the legend of Figure 1(b).

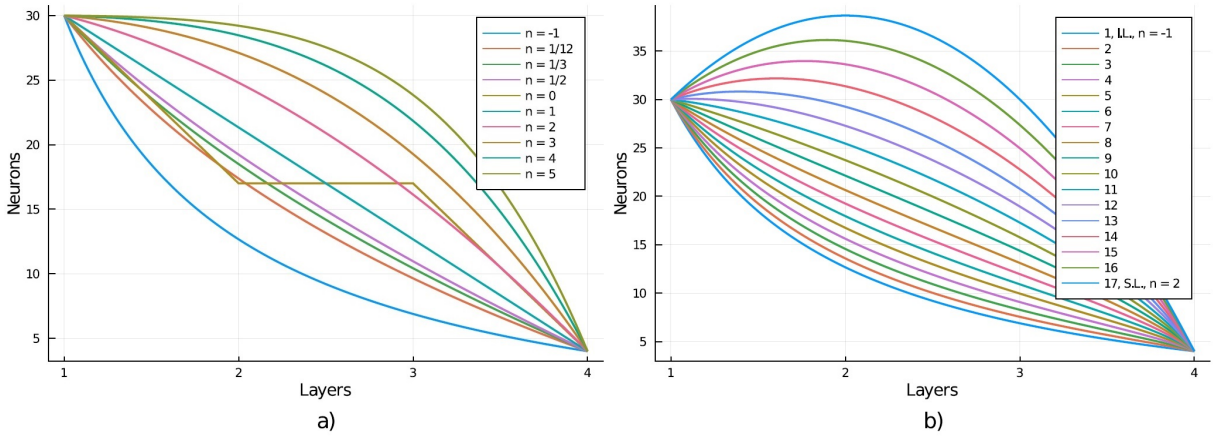

Figure 1. Curves of polynomials used for a neural network of 2 hidden layers and with 30 and 4 neurons in its input and outputs layers respectively.

Equation 2 and Equation 3 are inspired by the geometric concept of the bisection method, which splits an interval k times, and produces $2^k - 1$ line divisions inside the original interval. This concept is recollected here as the divisions are the intermediate curves, that are constrained to the boundary curves, inferior and superior limits. Thereby Eq. 2 and Eq. 3 have k in their indices, due to the fact that, at each splitting phase, the newly created curve is the product of its adjacent parameterizations, then the number of curves grows each time by 2^{k-1}. Therefore, the indice k is very important on both equations, Eq. 2 and Eq. 3 because two curves that lay at the vicinity of each other at one of these splitting iterations will not be next to each other at the subsequent iteration.

The availability of two methods of topology setting is proposed, one with each curve being separately parameterized with a given polynomial of the form of Eq. 1, another that can be subdivided into two cases, a nonlinear represented by Eq. 2 and a linear shown by Eq. 3, whose two boundary curves are parameterized and supplied to these equations to generate the intermediate curves, which can be modulated by the exponents f and g or by the coefficients v and w. And, common to both of them is the necessity of rounding the output of the curve at the position of the layer, in order to get a natural number. The rounding rules used were the standard and if the output were right in the middle of two integers, then the two cases would be tested. In this study, the feed-forward architectures were determined through the

first method, while the deep feedforward topologies were designed using the second technique, in particular with Eq. 2 and $f = g = 0.5$. When deep feedforward topologies with a single hidden layered were built, the number of neurons in this layer was increased one by one from 30 up to 70.

Nevertheless, the genetic idea behind the work presented by Völtz (2019) has not been left aside, but an evolutionary approach has been undertaken with the help of the validation data set. This concept of using this data set to optimize and tune hyperparameters is well known and described on Brownlee (2017), therefore, it has been employed to measure the accuracy of the calculated models with the parameterization approach. And, as an evolutionary technique indeed, the characteristics of the best performing models were kept on to propose other models.

3.5 Training, validation and test procedures

The methodology for these steps is tricky because underneath it lies a matter of accuracy and generalization trade-off since a model can be very accurate on a data set and very inaccurate on another, hence this network may have some accuracy but its generalization ability is poor. On the other hand, there might be a model which gives less accuracy on one or some data sets, however, is capable of generalizing because it can give most of the classifications right at a broad set of samples. Although there is indeed a better option, that consists of networks that respond with greater accuracy in all data sets. Then might be good if all trained models could be classified themselves in a rank of first, second and third class. Thereby, first-class stands for those networks that offer accuracy equal to 1 in all data sets (training, validation, and testing), second-class stands for the ones which give accuracy of 1 at two sets and a slightly smaller one on another, and, for the last, models that can't give accuracy equal to one in any data set.

The order of magnitude of the last value of the loss function during training gives a precise clue whether the accuracy in the training set is equal or very close to 1. Therefore, if this parameter matched the criterion for the given loss function, like at least an order of magnitude equal to -2 for cross-entropy, then the model would be applied on the validation data set, and, in case it achieved an accuracy of 100%, this model would be tried on the testing data set. Along with this natural selection-like lane, all data has been stored to be used by and for the criteria described in section 4.

Also each individual topology has been trained 500 times, so there has been created 500 models with the very same characteristics for each topology, each of those trainings will be denoted as *realizations*, owing to the fact that this term means when a model is fully trained and created using a topology and a set of fixed hyperparameters, and, thereby realization shall not be mistaken with the iterations of the optimization procedure inside the training of a single model. This procedure has been carried out since it was clear that the behavior of the phenomenon was probabilistic and, thus, the overall performance would be reliably qualified if there were as many training samples, that each parameter could converge steadily to a value or a distribution. At the beginning of this research just 100 of those realizations have been used, albeit with increasing this number, the convergence improved considerably until 500 realizations.

This approach reveals much of the intrinsic nature of each topology, since the hyperparameters are fixed among them, due to the fact that Julia Flux initiates the weights and biases randomly at every training and thus every realization has random entries. Then, if the output, the performance of the 500 models of the same topology for instance, may be classified and calculated into a probabilistic mass function (PMF) of errors (relative to the training data set) and another quantity of models in each accuracy level (relative to the validation data set), these results will be able to predict the response of a topology onto further use. This justifies the use of a greater number of realizations, like 500, in order to get more samples, which will lead to a more accurate PMF.

4. PERFORMANCE CRITERIA

4.1 Error at the end of the training process

The loss function indicates, regarding the method employed, how far the model is from the actual expected outputs during training, thus its last value informs how far the classification of the network of the training set is from the truth. Consequently, the very last output of the loss function, as introduced in subsection 3.5, may be used to infer whether a model has good accuracy and is thereby worth validation. Nevertheless, the curve that had been drawn during the optimization process is a source of information, such as which local minima the model is falling into, if the optimization is badly affected by a hyperparameter when compared to a smooth curve granted by another array of hyperparameters, or if the minimum was found earlier or later on the prescribed iterations.

4.2 Error distribution

During all the realizations for a single topology, the developed computational routine registered the order of magnitude using base 10 of the last value of the loss function. Therefore all the given errors relative to the training data set can be separated into bands, where each band comprehends an order of magnitude. However, this routine also recorded the quantities of models that were enclosed within each band, thereby at any realization along with the prescribed number

of them, there would be a record of how many models had fallen into a magnitude of error, and, dividing this quantity by the entire number of trained models, the importance of this error magnitude within the whole group of errors would be evaluated. But this importance, *i.e.*, this relative quantity is a function of the realization, since these quantities are changing. The curves the importance of each band makes along the realizations are of fundamental interest, since if those curves converge to a steady value, a probabilistic mass function (PMF) will appear. And after all the calculations, the result of those curves is a graph of the relative quantity of each band along with the realizations. All of those graphs converged to a probabilistic distribution, as shown in Figure 2, then the ordinate axis is limited between 0 and 1.

This distribution gives important information about the facility of training or fitting the model to the training data set. The greater percentage of the PMF the smallest errors take, the easier is to fit a topology into the training data set, and therefore the greater will be its accuracy. As the weight and biases initialization is randomly done, the curves exemplified in Figure 2 converged to the same distribution. What constructs a concrete way for predicting how difficult would it be to train a particular topology.

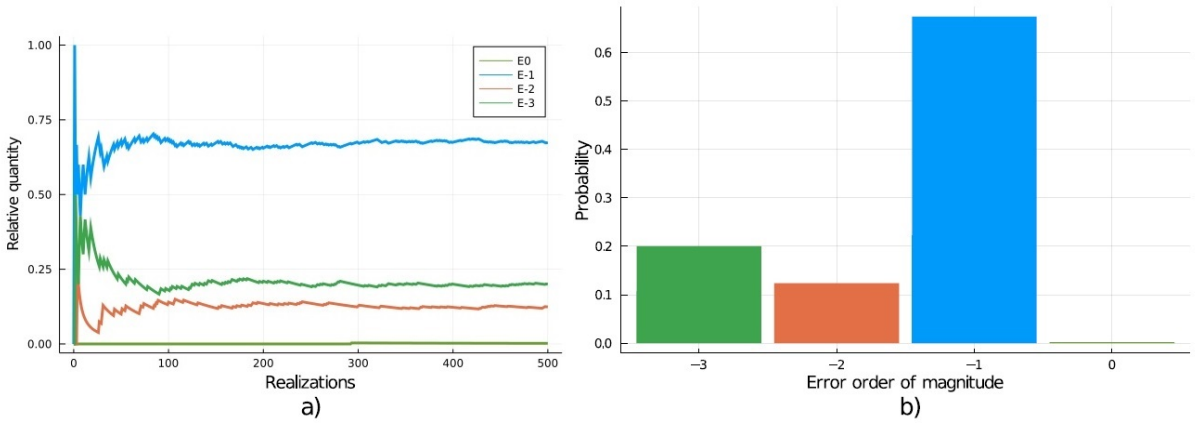

Figure 2. Quantity of each error magnitude relative to the quantity of all magnitudes along the 500 realizations. The magnitudes are presented on the legend at the top-left corner of the left graph as well as on the horizontal axis of the right graph, of the PMF. Illustration of distribution of topology 30/23/16/9/2.

4.3 Accuracy

The accuracy is the ratio of the number of right classifications by the number of all samples. Although this classification has to be evaluated on a particular data set (so a single model does have three accuracies) and there are 500 models for each topology, again a discrete probabilistic distribution appears. Thus, if the methodology followed to reach the error PMF is used again, then a new PMF will appear for the accuracy at a data set. Like for the error distribution, the quantity of trained models that achieve something (a level of accuracy for this criterion) is divided by the whole quantity of qualified models. And as shown both in Table. 3 and in Figure 3, there is a peak at an accuracy value of 0.64. One also can notice that this relative quantity or probability in the context of PMF decreases as it furthers in both directions away from the maxima location. Thus the interest shall lay on topologies that give more models with an accuracy of 1 or with the peak at the higher value of accuracy.

Table 3. Distribution of quantity of models in each accuracy value for topology 30/20/13/8/4

Accuracy	1.00	0.91	0.82	0.73	0.64	0.55	0.45	0.36	0.27	0.18	0.09	0.00
Quantity	1	5	19	34	46	31	19	9	3	1	0	0

5. RESULTS

Generally, topology parameterizations with higher n-degree led to better PMFs of accuracy at both data sets. Thus these PMFs presented peaks at higher accuracy values and more samples into the band of accuracy equal to 1. Also, topologies with this behaviour had their degree n steadily diminishing according to the rise of the number of hidden layers. There has been revealed peaks of quantity of good performing models on validation data set with one parameterization with upwards concavity and one with downwards concavity, however, the last one has brought better results. Although the far best results have been given by the single hidden layer topologies, especially at topologies 30/42/4 (4 models with an accuracy of 1 at the validation data set) and 30/40/2 (12 of these optimum models). On the test data set, topology 30/53/2 gave 2 optimum models (accuracy equal to 1) out of its 9 optimum models on the validation data set, while the best

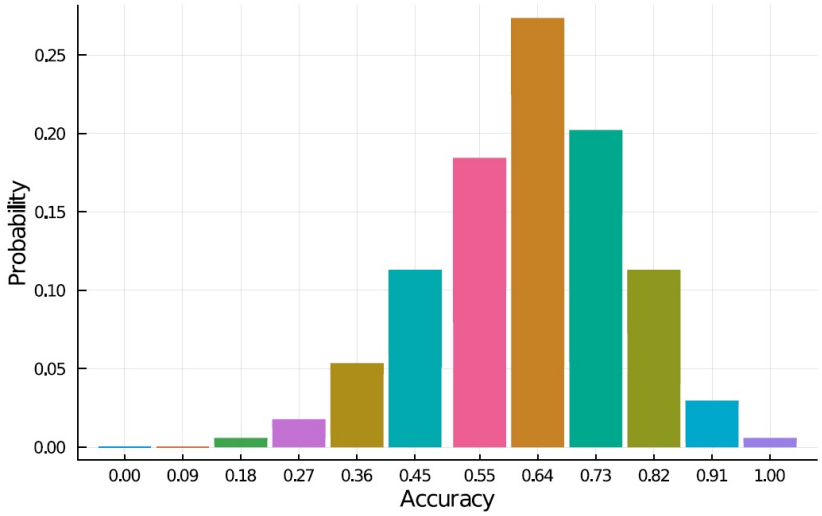

Figure 3. Distribution of relative quantity of models in each accuracy value (PMF) for topology 30/20/13/8/4.

performing topology with 4 outputs is still 30/42/4, with half of the optimum models showing 0.71 accuracies on the test data set and a model with accuracy equal to 1 at the test set and 0.82 at the validation data set. The better performance of the single hidden layer topologies is explained by the ease in which the gradient is calculated and propagated throughout the architecture during training, therefore the problem of gradient vanishing is much lesser common, while the greater quantity of neurons in the middle layer reduces the risk of overfitting, that is often seen on feed-forward topologies of this kind. As expected, the error distribution for those deep feedforward single hidden layer networks is dominated by errors of very small orders of magnitude.

The number of optimum models of the deep feedforward 3-layered topologies varied cyclic along with the escalate in the number of neurons in the intermediate layer, peaking and diminishing. But the amplitude of the peaks increased steadily to the best performing topology and then slowly dropped. Therefore marking where the best option might be, although it may occur, as it did, that a topology gives more optimum models at the validation data set context but fewer compared to another topology qualified on the test data set. Situation exemplified by 30/40/2 (12 optimum models at the validation phase and none of them at the test) and 30/53/2 (9 optimum models at validation and 2 at test) respectively. Also illustrated with 4-output architectures by 30/42/4, 30/59/4 and 30/64/4, since the first one had only 4 optimum models at validation set but 1 model with 100% of accuracy at test data set, while 30/59/4 and 30/64/4 had 5 optimum models at the validation phase but none at the testing step.

Since some of the optimum models on the validation data set do not perform with the same quality over the test set, the criterion for segregation of models on the validation phase was shrunk to 0.82 of accuracy, thus a model with this accuracy is taken to be qualified on the test phase. Therefore there may be found models that have accuracy equal to 1 at the test data set and less at the validation one. It has to be settled down then a way to discern whether a model with an accuracy of 100% at validation is better than one of this level of accuracy at test data set, and preference has to be given to the best performing model on the broadest set or on the set with greater variance, due to generalization capability being an important quest. Hence, a rank among models can be made by giving each one of them the following score:

$$s = \sum^{N} A_i Q_i \tag{4}$$

where A_i is the accuracy at the i-th data set, Q_i is the quantity of samples in the i-th set, and N is the number of sets

6. CONCLUSIONS

For this kind of application and particular data type, the best option lies on shorter, with only one hidden layer, architectures trained with cross-entropy loss function and bound together with sigmoid activation function and softmax before the last (output) layer, and without any mainstream regularization technique. This study has demonstrated and illustrated the probabilistic nature of the phenomena, which sets light on the search for a particular family of topologies for a classification problem alike. In truth, the probabilistic distribution or error order of magnitude at the training data set shall be used even to detect the occurrence of under or overfitting. While the PMF of the accuracies can be used to quantify the generalization capability of a topology, even though there has to be a weighting on the PMFs generated at the validation and test data sets, as shown in Eq. 4.

This research also lays a foundation for a methodology that enables the detection of patterns regarding the cause and effect relation between architecture and performance, through the polynomial approach, as well as through the parame-

terization of deep feedforward topologies. The parameterization techniques and the procedure of the realizations are easy to program and implement on an application scale, such as in the industrial sector. As shown on the graphs and tables, the generated probability mass functions for order of magnitude of the error and accuracy can be used to asses performance.

The parameterization techniques detailed in Eq. 1-3 can track topologies within and beyond the rectangle interval of feed-forward networks, and they may use geometrical concepts, such as concavity and smoothness, to trace curves that will lead to customized performance. As explained prior, the last two techniques may be also used to combine curves to hybridize and geometrically crossover their characteristics, which can lead to the crossover of performance capabilities.

This manuscript has demonstrated a good path towards building and selecting reliable and accurate neural networks for the detection of damage in composite beams. Although the best results were obtained to the first way of classification. As such, the application and relevancy of all this acquired knowledge are justified and affirmed. Therefore the use of those techniques is recommended for the classification of whether an item is damaged or not, and that is of fundamental interest since the input data of the trained and tested neural network is generated by non-destructive vibration experiments, which are quick and relatively cheap to perform, even *in situ*.

7. ACKNOWLEDGEMENTS

The authors acknowledge the financial support of the State Research Founding Agency (FAPESC process number: 2017TR1747, 2017TR784, 2019TR779). As well as, Coordination for the Improvement of the Higher Level Personnel (CAPES), National Council for Scientific and Technological Development (CNPq).

8. REFERENCES

Adams, R., Walton, D., Flitcroft, J. and Short, D., 1975. "Vibration testing as a nondestructive test tool for composite materials". In *Composite reliability*, ASTM International.

Babu, J., Sunny, T., Paul, N.A., Mohan, K.P., Philip, J. and Davim, J.P., 2016. "Assessment of delamination in composite materials: a review". *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 230, No. 11, pp. 1990–2003.

Bandara, R.P., Chan, T.H. and Thambiratnam, D.P., 2014. "Structural damage detection method using frequency response functions". *Structural Health Monitoring*, Vol. 13, No. 4, pp. 418–429.

Bishop, C.M. *et al.*, 1995. *Neural networks for pattern recognition*. Oxford university press.

Brownlee, J., 2017. "What is the difference between test and validation datasets?" https://machinelearningmastery.com/difference-test-validation-datasets/. Acessed 5 March 2021.

Campbell, F.C., 2010. *Structural composite materials*. ASM international.

De Medeiros, R., Vandepitte, D. and Tita, V., 2018. "Structural health monitoring for impact damaged composite: a new methodology based on a combination of techniques". *Structural Health Monitoring*, Vol. 17, No. 2, pp. 185–200.

Fan, W. and Qiao, P., 2011. "Vibration-based damage identification methods: a review and comparative study". *Structural health monitoring*, Vol. 10, No. 1, pp. 83–111.

Fang, X., Luo, H. and Tang, J., 2005. "Structural damage detection using neural network with learning rate improvement". *Computers & structures*, Vol. 83, No. 25-26, pp. 2150–2161.

Farrar, C.R. and Worden, K., 2012. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons.

Garcia, D. and Trendafilova, I., 2014. "A multivariate data analysis approach towards vibration analysis and vibration-based damage assessment:: Application for delamination detection in a composite beam". *Journal of Sound and Vibration*, Vol. 333, No. 25, pp. 7036–7050.

Gomes, G.F., de Almeida, F.A., Junqueira, D.M., da Cunha Jr, S.S. and Ancelotti Jr, A.C., 2019a. "Optimized damage identification in cfrp plates by reduced mode shapes and ga-ann methods". *Engineering Structures*, Vol. 181, pp. 111–123.

Gomes, G.F., Mendez, Y.A.D., Alexandrino, P.d.S.L., da Cunha, S.S. and Ancelotti, A.C., 2019b. "A review of vibration based inverse methods for damage detection and identification in mechanical structures using optimization algorithms and ann". *Archives of computational methods in engineering*, Vol. 26, No. 4, pp. 883–897.

Gul, M. and Catbas, F.N., 2009. "Statistical pattern recognition for structural health monitoring using time series modeling: Theory and experimental verifications". *Mechanical Systems and Signal Processing*, Vol. 23, No. 7, pp. 2192–2204.

Junior, R.F.R., de Almeida, F.A. and Gomes, G.F., 2020. "Fault classification in three-phase motors based on vibration signal analysis and artificial neural networks". *Neural Computing and Applications*, Vol. 32, No. 18, pp. 15171–15189.

Karpathy, A., 2017. "The vanishing gradient problem". https://ayearofai.com/rohan-4-the-vanishing-gradient-problem-ec68f76ffb9b. Acessed 23 November 2018.

Karsh, P., Mukhopadhyay, T. and Dey, S., 2018. "Spatial vulnerability analysis for the first ply failure strength of composite laminates including effect of delamination". *Composite Structures*, Vol. 184, pp. 554–567.

Khan, A., Ko, D.K., Lim, S.C. and Kim, H.S., 2019. "Structural vibration-based classification and prediction of delami-

nation in smart composite laminates using deep learning neural network". *Composites Part B: Engineering*, Vol. 161, pp. 586–594.

Looney, C.G., 1996. "Advances in feedforward neural networks: demystifying knowledge acquiring black boxes". In *IEEE Transactions on Knowledge and Data Engineering*. IEEE.

Ng, A., 2018. "Principle component analysis algorithm". https://www.coursera.org/lecture/machine-learning/principal-component-analysis-algorithm-ZYIPa. Acessed 10 January 2019.

Oliver, G.A., Ancelotti, A.C. and Gomes, G.F., 2021. "Neural network-based damage identification in composite laminated plates using frequency shifts". *Neural Computing and Applications*, Vol. 33, No. 8, pp. 3183–3194.

Pereira, J.L.J., Chuman, M., Cunha Jr, S.S. and Gomes, G.F., 2020. "Lichtenberg optimization algorithm applied to crack tip identification in thin plate-like structures". *Engineering Computations*.

Pereira, M.S. and Bezerra, E.M., 2007. "Utilization of neural networks in prediction of aeronautical compounds behaviour under shearing (in portuguese)". In *Meeting of cientific iniciation and post-graduate program of ITA (in Portuguese)*. São José dos Campos, SP, Brazil.

Reis, P.A., 2020. *Artificial neural networks applied to structural damage identification using dynamic response and signal processing*. Master's thesis, Mechanical Engineering Graduate Program, Santa Catarina State University, Joinville, Brasil.

Reis, P.A., Iwasaki, K.M.K., Voltz, L.R., Cardoso, E.L. and De Medeiros, R., 2021. "Damage detection of composite beams structures using vibration response and artificial neural networks". In *Proceedings of the 5th Brazilian Conference on Composite Materials - BCCM5*.

Reis, P.A., Voltz, L.R., Cardoso, E.L. and De Medeiros, R., 2019. "Damage identification on composite beams using computational vibration model with uncertainties and artificial neural networks". In *Proceedings of the 25nd International Congress of Mechanical Engineering - COBEM 2019*.

Silva, I., Spatti, D.H. and Flauzino, R.A., 2010. *Artificial neural networks for engineering and applied sciences (in Portuguese)*. Artliber, São Paulo.

Valdes, S.D. and Soutis, C., 1999. "Delamination detection in composite laminates from variations of their modal characteristics". *Journal of sound and vibration*, Vol. 228, No. 1, pp. 1–9.

Vidhya, T.A., 2016. "A practical guide to principle component analysis (pca) in r and python". https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/comment-129382. Acessed 10 December 2018.

Völtz, L.R., 2019. *Fault diagnosis in composite structures using artificial neural network and principal component analysis*. Master's thesis, Mechanical Engineering Graduate Program, Santa Catarina State University, Joinville, Brasil.

Völtz, L.R., Cardoso, E.L. and De Medeiros, R., 2017. "Damage detection on aluminum beams using vibration-based method and artificial neural networks". In *Proceedings of the 24nd International Congress of Mechanical Engineering - COBEM 2017*.

Völtz, L.R., Cardoso, E.L. and De Medeiros, R., 2018. "Artificial neural networks applying to detect damage in carbon fiber/epoxy composites". In *Proceedings of the 4th Brazilian Conference on Composite Materials - BCCM4*.

Völtz, L.R., Cardoso, E.L. and De Medeiros, R., 2019. "Structural health monitoring in mechanical systems from changes in their vibration characteristics based on artificial neural networks and pca". In *Proceedings of the 25nd International Congress of Mechanical Engineering - COBEM 2019*.

Yam, L., Yan, Y. and Jiang, J., 2003. "Vibration-based damage detection for composite structures using wavelet transform and neural network identification". *Composite Structures*, Vol. 60, No. 4, pp. 403–412.

Zhang, Z. and Friedrich, K., 2003. "Artificial neural networks applied to polymer composites: a review". *Composites Science and technology*, Vol. 63, No. 14, pp. 2029–2044.

Zhang, Z., He, M., Liu, A., Singh, H.K., Ramakrishnan, K.R., Hui, D., Shankar, K. and Morozov, E.V., 2018. "Vibration-based assessment of delaminations in frp composite plates". *Composites Part B: Engineering*, Vol. 144, pp. 254–266.

Zhang, Z., Shankar, K., Morozov, E.V. and Tahtali, M., 2016. "Vibration-based delamination detection in composite beams through frequency changes". *Journal of Vibration and Control*, Vol. 22, No. 2, pp. 496–512.

Zhang, Z., Shankar, K., Ray, T., Morozov, E.V. and Tahtali, M., 2013. "Vibration-based inverse algorithms for detection of delamination in composites". *Composite Structures*, Vol. 102, pp. 226–236.

Zheng, T., Luo, W., Tong, H. and Liang, X., 2021. "A two-step method for delamination detection in composite laminates using experience-based learning algorithm". *Structural Health Monitoring*, p. 14759217211018114.

Zou, Y., Tong, L. and Steven, G.P., 2000. "Vibration-based model-dependent damage (delamination) identification and health monitoring for composite structures—a review". *Journal of Sound and vibration*, Vol. 230, No. 2, pp. 357–378.

9. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.