# A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks

Long Viet Ho[1,2], Duong Huong Nguyen[1,3], Mohsen Mousavi[4], Guido De Roeck[5], Thanh Bui-Tien[6], Amir H. Gandomi[4], Magd Abdel Wahab[7,1*]

[1]*Soete Laboratory, Department of Electrical Energy, Metals, Mechanical Constructions and Systems, Faculty of Engineering and Architecture, Ghent University, 9000 Gent, Belgium; HoViet.Long@ugent.be (L.V.H.); huongduong.nguyen@ugent.be (D.H.N);*

[2] *Department of Bridge and Tunnel Engineering, Faculty of Civil engineering, Campus in Ho Chi Minh City, University of Transport and Communications, Ho Chi Minh 700000, Vietnam; longhv_ph@utc.edu.vn*

[3]*Department of Bridge and Tunnel Engineering, Faculty of Bridge and Road, National University of Civil Engineering, Hanoi 100000, Vietnam; duongnh2@nuce.edu.vn*

[4]*Faculty of Engineering and IT, University of Technology Sydney, Ultimo, NSW 2007, Australia; mohsen.mousavi@uts.edu.au, gandomi@uts.edu.au*

[5]*Department of Civil Engineering, Structural Mechanics, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium; guido.deroeck@kuleuven.be*

[6] *Department of Bridge and Tunnel Engineering, Faculty of Civil engineering, University of Transport and Communications, Hanoi 100000, Vietnam; btthanh@utc.edu.vn*

[7]*CIRTech Institute, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh City, Vietnam*

**Corresponding author:* magd.a.w@hutech.edu.vn; magd.abdelwahab@ugent.be

**Abstract:** Finite element (FE) based structural health monitoring (SHM) algorithms seek to update structural damage indices through solving an optimisation problem in which the difference between the response of the real structure and a corresponding FE model to some excitation force is minimised. These techniques, therefore, exploit advanced optimisation algorithms to alleviate errors stemming from the lack of information or the use of highly noisy measured responses. This study proposes an effective approach for damage detection by using a recently developed novel swarm intelligence algorithm, i.e. the marine predator algorithm (MPA). In the proposed approach, optimal foraging strategy and marine memory are employed to improve the learning ability of feedforward neural networks. After training, the hybrid feedforward neural networks and marine predator algorithm, MPAFNN, produces the best combination of connection weights and biases. These weights and biases then are re-input to the networks for prediction. Firstly, the classification capability of the proposed algorithm is investigated in comparison with some well-known optimization algorithms such as particle swarm optimization (PSO), gravitational search algorithm (GSA), hybrid particle swarm optimization-gravitational search algorithm (PSOGSA), and grey wolf optimizer (GWO) via four classification benchmark problems. The superior and stable performance of MPAFNN proves its effectiveness. Then, the proposed method is applied for damage identification of three numerical models, i.e. a simply supported beam, a two-span continuous beam, and a laboratory free-free beam by using modal flexibility indices. The obtained results reveal the feasibility of the proposed approach in damage identification not only for different structures with single damage and multiple damage, but also considering noise effect.

***Keywords:*** Hybrid approach, marine predator algorithm-feedforward neural networks (MPAFNN), vibration experiment, damage detection, modal flexibility index.

## I. Introduction

Structures under service will inevitably undergo some damage due to their permanent exposure to operational loads, environmental effects or accidental events. Maintenance and repair of existing bridge structures have gained more and more attention recently. To achieve an effective, low-cost, and timely manner SHM, damage detection must be conducted at an early stage. In traditional assessment methods, visual inspection techniques play a significant role in collecting data on in-service bridges. The collected data are further processed to assess the health condition of structures. However, there are some limitations involved with these techniques such as the existence of invisible or inside structural damage or inequalities of the inspector's competences.

In order to overcome these shortcomings, many researchers successfully applied a physical model-based method for damage identification. Using the direct changes in natural frequencies or displacement mode shapes between intact and damaged conditions, they identify the location and the level of damage. However, this approach can use an optimization algorithm to solve the model updating problem [1-6]. Therefore, these approaches can become time-consuming in case of the occurrence of multiple damage scenarios or complex structures. The authors in [7] also used the inverse problem-based approach for damage detection. A regularized level set method was used to identify defects in a piezoelectric material via an iterative procedure. The obtained results confirmed that the proposed algorithm could successfully determine the number, approximate location and shape of defects in the piezoelectric domain.

In contrast, modal-based damage detection methods are another set of techniques that have been successfully applied to identify defects without iteration. These approaches use damage sensitive indices to detect, localize and evaluate damage in civil engineering structures. The first and simplest index based on displacement mode shapes e.g. Modal Assurance Criterion (MAC) or Coordinate Modal Assurance Criterion (COMAC) was used in studies [8-10]. Although the obtained results indicated the high potential of these indices in detecting the presence of damage, their capability in localizing damage showed some limitations. Modal strain energy-based (MSE) methods are another set of modal-based techniques which have received many positive results in damage detection and localization. Many numerical studies confirmed high accuracy and reliability of these methods in detecting and locating damage at different noise levels [11-16]. However, directly extracting modal strain energy indices from measurements is still a challenge. Modal curvature method (MCM) is another popular mode shape-based approach for damage identification. Wahab and De Roeck [17] applied modal curvature (MC) and curvature damage factor (CDF) to both simulated data and a real case, the bridge Z24. They found that MC was more precise when using the lower modes. Performance of higher modes MC could be guaranteed by increasing the measurement grid. They emphasized that when severe damage occurred in structures, CDF revealed its supremacy of damage localization. Authors in [18-20] recommended using modal curvature method (MCM) for localization of damage. However, this method is not a suitable choice, especially in large sensor spacing condition or when only higher modes could be extracted. Moreover, the normalization of two-mode shapes of healthy and damaged states is necessary to guarantee the quality of damage identification.

Modal flexibility method (MFM) combines natural frequencies and displacement mode shapes for determining damage in structures. One of the advantages of this method is that it can indicate the damage position based on the first few lower modes. In other words, the flexibility index is sensitive to changes in stiffness of the structure even when only lower modes are used. Besides, normalization of mode shapes is unnecessary for increasing the accuracy of defect localization. MFM was applied successfully in [21-23]. However, the quantification of damage level was not mentioned in these studies. In reality, natural frequencies/lower-mode is easier to identify than mode shapes/higher-mode. Therefore, the real application of this approach is feasible due to its simplicity and low cost. This study, therefore, exploits MFM to develop a tool for SHM of large-scale structures. The limitation of this approach in damage quantification is improved by using neural networks (NNs). There are several NNs e.g. recurrent neural network (RNN), spiking neural network (SNN), Feedforward neural network (FNN), etc. This study focuses on FNN due to its simplicity in information transfer from inputs to predict outputs.

Machine learning algorithms have shown capability in solving complicated problems, like classification, regression and clustering, etc. A concerning crucial matter, in all neural networks, is the learning process. Learning from experience or available data is the key that helps NNs to overcome complex problems. Especially, in applications where the approach has to deal with big data. This approach uses two categories of learning techniques: unsupervised and supervised. The former is only based on input data to interpret and group data. The latter, however, uses both input and output data to develop a predictive model. While unsupervised learning is used in clustering, supervised learning is used in classification and regression. For the high performance of a trained neural network (NN), a learning method or trainer is involved during the learning process of a NN. Stochastic and deterministic methods are two major categories that can be applied to that purpose. Less computational time and simplicity are strong points of the deterministic method where a popular gradient-based algorithm, namely back-propagation (BP), serves as a training method. Reducing computational time is always a target that many researchers aim to. Authors in [24-25] proposed novel approaches. In these approaches, they did not need to use a classical discretization such as FEM. After the network has been trained, solutions were obtained extremely fast based on collocation strategy. They could deal with the forward and inverse/optimization problem in the same way (Deep neural networks DNN and no FEM at all) and automatically account for uncertainties. However, the initial values of connection weights, biases, learning rate, and momentum have significant effects on the convergence. For this reason, the use of the BP algorithm in the training process can result in a mistaken point or converging to local minima rather than to the global minimum. To reduce the probability of local optima avoidance, a stochastic trainer based on the optimization strategy for training can be used. In the second approach, random solutions are generated in the initial iterations, and then they are evolved as iteration pass. Although the convergence rate of the stochastic approach is slower than that of the deterministic one, it owns high potential in local solutions' avoidance. The kernel of the optimization-based stochastic method is a heuristic optimization algorithm. The well-developed optimization field provides a strong foundation for the development of this approach. Many well-known algorithms are capable of global optimal search e.g. genetic algorithm GA, particle swarm optimization (PSO), improved particle swarm (IPSO) [26], orthogonal diagonalization-improved particle swarm optimization (IPSOOD) [27], gravitational search algorithm (GSA), simulated annealing (SA), grey wolf optimizer (GWO), water cycle algorithm (WCA) [28], etc. Water cycle algorithm (WCA), a novel optimization method, is inspired by the hydrologic cycle process in nature. WCA focuses on the motion of raindrop from the atmosphere to the streams, then to rivers that flow to the sea or from the atmosphere directly to the sea. Random raindrops represent candidate solutions. Many good raindrops are considered as a river while the rest of the raindrops represent streams. All streams and rivers wind up in the sea. The sea indicates the best raindrop (the best optimal point). Evaporation and raining condition help WCA to overcome the local stagnation. Four constrained optimization problems and seven engineering problems were used to validate the performance of WCA. The obtained results confirmed its effectiveness via computational cost and accuracy of solutions. Some of which were applied successfully in training feedforward neural network (FNN).

Recently, a new nature-inspired heuristic optimization algorithm, namely marine predator algorithm proved its prominent performances over many other well-known algorithms regarding exploration and exploitation of engineering problems [29]. Therefore, the authors proposed a new stochastic approach based on optimization strategy in training FNN, namely marine predator FNN (MPAFNN). The study aims to evaluate the capability of the proposed approach in escaping from local optimal solutions and improving the accuracy of prediction in structural engineering problems. The proposed FNN is fed by modal flexibility index and uses a supervised technique for learning.

## II. Methodology of the proposed approach

### 2.1 Modal flexibility indices

Modal flexibility matrix is defined as an inverse of the stiffness matrix [30]. The presence of damage in a structure causes a reduction of stiffness. It results in a rise in the flexibility of structures when a small failure occurs. In other words, the presence of a defect can cause a more flexible zone in the vicinity of the damage excepting the particular location such as the fixed end of a cantilever beam. Therefore, these changes in the observed flexibility of the structure can be used as a damage indication. The indicator can be calculated using vibration properties such as natural frequency values and mass-normalized displacement mode shapes. Generally speaking, a comparison between two flexibility matrices extracted from two sets of dynamic characteristics is the working principle of the method. For instance, at a given location $s$ and number of considered modes $nm$, a modal flexibility element $MF_s$ of the modal flexibility matrix of a tested structure can be identified:

$$MF_s = \sum_{i=1}^{nm} \frac{1}{f_i^2} \Phi_{si} . \Phi_{si}^T \tag{1}$$

Where $i$ implies mode number, $f_i$ denotes natural frequency of mode $i$, $\Phi_{si}$ are mass-normalized mode shapes at location $s$ of mode $i$, superscript $^{T}$ indicates "Transpose". The interesting point in (1) is the inverse relationship between the modal flexibility and the square of frequencies. This implies that lower frequencies have a higher contribution to modal flexibility. It makes modal flexibility efficient when only a few lower frequencies are identified experimentally. In real applications, the lower modes of a real structure are often easier to measure compared to the higher ones. Therefore, this index is intrinsically interesting because of its feasibility for practical applications.

For damage identification, two modal flexibility matrices obtained from healthy and unhealthy (damaged) states have to be determined. Therefore, a set of modal properties of the intact structure is used to calculate the flexibility matrix as:

$$MF_H = [MF_s]_{\mathrm{H}} = \left[ \sum_{i=1}^{nm} \frac{1}{f_i^2} \Phi_{si} . \Phi_{si}^T \right]_H \tag{2}$$

Then another set of dynamic characteristics of the damaged structure is determined as:

$$MF_D = [MF_s]_D = \left[ \sum_{i=1}^{nm} \frac{1}{f_i^2} \Phi_{si} . \Phi_{si}^T \right]_D \tag{3}$$

A symmetric matrix of indicators for damage identification can be computed by using the difference between the two obtained matrices from (2) and (3):

$$\Delta MF = MF_D - MF_H = [MF_s]_D - [MF_s]_H \tag{4}$$

Each column in the indicator matrix $\Delta MF$ represents the measurement points on the structure. To identify the points (degrees of freedom, DOFs) on the structure which are influenced by damages, the maximum absolute value of each column is determined. Then, these values are used to indicate the presence and positions of defects. However, for a clearer view of defect location, especially in the case of multiple damage scenarios, the absolute value of curvatures is calculated using equation (5). The indices are named modal flexibility-based curvature *MFC* [31] to distinguish from the damage indices using the flexibility changes.

$$MFC_i = \left| \frac{\Delta MF_{i-1} - 2.\Delta MF_i + \Delta MF_{i+1}}{(\Delta s)^2} \right| \tag{5}$$

Where $i$ indicates the $i^{th}$ DOF, $\Delta s$ denotes the distance between DOFs (measurement points) on the structure. Details of damage identification using *MFC* are introduced in sections to follow.

There is a challenge in real applications that is the mass-normalized mode shape can be obtained when an input-output measurement is conducted. For a laboratory beam, the impact hammer is used to generate excitation at each position of sensor in the vertical direction. In other words, the hammer provides pulse excitation as inputs, whereas outputs are acceleration obtained by sensors. This study only focuses on investigating the potential of *MFC* index in damage detection using the proposed hybrid computational intelligence approach instead of identifying the measured flexibility.

## 2.2 Marine predator algorithm – MPA

Marine predator algorithm (MPA) is a nature-inspired optimization method [29]. The basic idea of MPA is based on a flexible swap between two foraging strategies, e.g. Brownian and Lévy movements. This tradeoff aims to reach an optimal foraging strategy for predators. In other words, the combination of two foraging strategies increases the encounter rate between prey and predator in the marine ecosystem.

Previous studies indicated that, throughout the lifetime, a predator shows an equal percentage between Lévy and Brownian movement. However, it is very interesting that the speed ratio of prey to predator has a significant effect on the foraging strategy of the predator. In other words, depending on this ratio of velocity, the predator can move in Lévy flight or Brownian motion. High-velocity ratio, unit velocity ratio, and low-velocity ratio are three typical ratios in a marine ecosystem. Therefore, the flexible combination of these two movements will provide the optimal strategy of movement of a predator. Moreover, predators sometimes perform sudden, long, and vertical jumps when they face environmental problems such as the formation of eddy or human's activities e.g. fish aggregating device (FADs). The observed action may imply that predators are aiming for high potential in order to find a food-abundant environment. Preserving the location of successful foraging in their memory also allows predators to survive and thrive.

As the discussion, the main content of the MPA algorithm revolves around the relationship between predator and preys. However, preys also seek their food. It means that the preys then become predators. Therefore, the algorithm shows the relationship between the top predator (also known as Elite) and the prey. To initiate the MPA optimization process two same dimension matrices for Elite and Prey are constructed based on the population of search agents $p$, and the number of updating parameters $u$. Firstly, a prey matrix is constructed to contain the initial preys.

$$Prey = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,u} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,u} \\ \vdots & \vdots & \vdots & \vdots \\ X_{p,1} & X_{p,2} & \cdots & X_{p,u} \end{bmatrix}_{p \times u} \tag{6}$$

A top predator vector represents the fittest solution obtained from the prey matrix that is identified as:

$$Predator_{top} = \begin{bmatrix} X_1^T & X_2^T & \dots & X_u^T \end{bmatrix}_{1\times u} \tag{7}$$

Then Elite matrix is built by replicating $p$ times the top predator vector. The Elite matrix is used to update the better values of the top predator.

$$Elite = \begin{bmatrix} predator_{top} \\ predator_{top} \\ \vdots \\ predator_{top} \end{bmatrix}_{p\times 1} = \begin{bmatrix} X_{1,1}^T & X_{1,2}^T & \cdots & X_{1,u}^T \\ X_{2,1}^T & X_{2,2}^T & \cdots & X_{2,u}^T \\ \vdots & \vdots & \vdots & \vdots \\ X_{p,1}^T & X_{p,2}^T & \cdots & X_{p,u}^T \end{bmatrix}_{p\times u} \tag{8}$$

Same as other optimization algorithms, MPA also focuses on exploration in the initial loops and exploitation in the last loops. To do this, MPA relies on velocity ratio between predators and preys, the parameter that significantly affects the movement strategy of the predators, as mentioned above. Therefore, the ability of MPA in exploration and exploitation is exhibited in three main phases. One-third of the total number of iterations is used in each phase.

Phase 1: The current iteration $iter_{current} \le \frac{1}{3} Iter_{\max}$ , high-velocity ratio, $r_{velocity} \ge 10$, exploration stage. The prey's movement is faster than that of a predator. Therefore, the optimal strategy for a predator is standing still. Meanwhile, the prey can move in Lévy or Brownian, and it moves forward to the predator. This phase is interpreted in a mathematical model:

$$\overrightarrow{step}_{(i,j)} = \vec{R}_{B(i,j)} \times \left( \overrightarrow{\mathrm{Elite}}_{(\mathrm{i,j})} - \vec{R}_{B(i,j)} \times \overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} \right) \tag{9}$$

$$\overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} = \overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} + P \times R \times \overrightarrow{\mathrm{step}}_{(\mathrm{i,j})} \tag{10}$$

Where, $\vec{R}_B$ denotes Brownian random number vector; subscripts $i = 1, 2, \dots, p$ and $j = 1, 2, \dots, u$, are the number of population and variables, respectively, constant number $P$=0.5, $R \in [0,1]$ is a uniformly distributed random number in an interval from 0 to 1.

Phase 2: The current iteration $\frac{1}{3} Iter_{\max} < iter_{current} \le \frac{2}{3} Iter_{\max}$ , unit velocity ratio, $r_{velocity} = 1$, intermediate stage, converting from exploration to exploitation. Hence, the population is divided into two halves. One half is used for exploration while the other for exploitation. In the first half population, the motion in Lévy of the prey is considered as exploration capability. In the second one, the predator performs the Brownian motion. A new position of the prey is updated based on the predator's movement. In other words, exploitation capability mainly depends on the movement of the predator. The mathematical model of the second phase is performed via two steps.

Step 1: A half of population for exploration, $i$ from 1 to $p/2$

$$\overrightarrow{step}_{(i,j)} = \vec{R}_{L(i,j)} \times \left( \overrightarrow{\mathrm{Elite}}_{(\mathrm{i,j})} - \vec{R}_{L(i,j)} \times \overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} \right) \tag{11}$$

$$\overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} = \overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} + P \times rand \times \overrightarrow{\mathrm{step}}_{(\mathrm{i,j})} \tag{12}$$

Where $rand$ implies uniform random numbers in a range of 0 to 1, $\vec{R}_L$ denotes the Levy random number vector. This vector represents Lévy movement and is calculated as:

$$\vec{R}_L = 0.05 \times \frac{a}{|b|^{1/\alpha}} \tag{13}$$

Two normal distribution variables $a$ *and* $b$ are calculated by using standard deviations $\sigma_a$ *and* $\sigma_b$ and gamma function $\Gamma$ as follows:

$$a = Normal\left(0,\sigma_a^2\right), \text{ with } \sigma_a = \left[\frac{\Gamma(1+\alpha)\times\sin(\pi\alpha/2)}{\Gamma\left(\frac{1+\alpha}{2}\right)\times\alpha\times 2^{(\alpha-1)/2}}\right]^{1/\alpha} \text{ and } \alpha=1.5 \tag{14}$$

$$b = Normal\left(0,\sigma_b^2\right), \text{ with } \sigma_b=1 \tag{15}$$

Step 2: The other half of population for exploitation, $i$ from $p/2$ to $p$

$$\overrightarrow{step}_{(i,j)} = \vec{R}_{B(i,j)} \times \left(\vec{R}_{B(i,j)} \times \overrightarrow{\mathrm{Elite}}_{(\mathrm{i,j})} - \overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})}\right) \tag{16}$$

$$\overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} = \overrightarrow{Elite}_{(\mathrm{i,j})} + P \times CF \times \overrightarrow{\mathrm{step}}_{(\mathrm{i,j})} \tag{17}$$

The step size for movement of predator is adapted by using factor $CF$:

$$CF = \left(1 - \frac{iter_{current}}{Iter_{\max}}\right)^{\left(2\times\frac{iter_{current}}{Iter_{\max}}\right)} \tag{18}$$

Phase 3: The current iteration $\frac{2}{3} Iter_{\max} < iter_{current} \leq Iter_{\max}$, low-velocity ratio, $r_{velocity} = 0.1$, exploitation stage. This stage mainly focuses on exploitation. The movement of the predator is simulated in Lévy strategy. The position of prey is updated based on the predator's motion.

$$\overrightarrow{step}_{(i,j)} = \vec{R}_{L(i,j)} \times \left(\vec{R}_{L(i,j)} \times \overrightarrow{\mathrm{Elite}}_{(\mathrm{i,j})} - \overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})}\right) \tag{19}$$

$$\overrightarrow{\mathrm{Prey}}_{(\mathrm{i,j})} = \overrightarrow{Elite}_{(\mathrm{i,j})} + P \times CF \times \overrightarrow{\mathrm{step}}_{(\mathrm{i,j})} \tag{20}$$

The summary of the progression of the algorithm through the key parameters, such as the velocity ratio, the best strategy for predator and prey, and the number of the population associated with the corresponding phase, is depicted in Fig. 1.

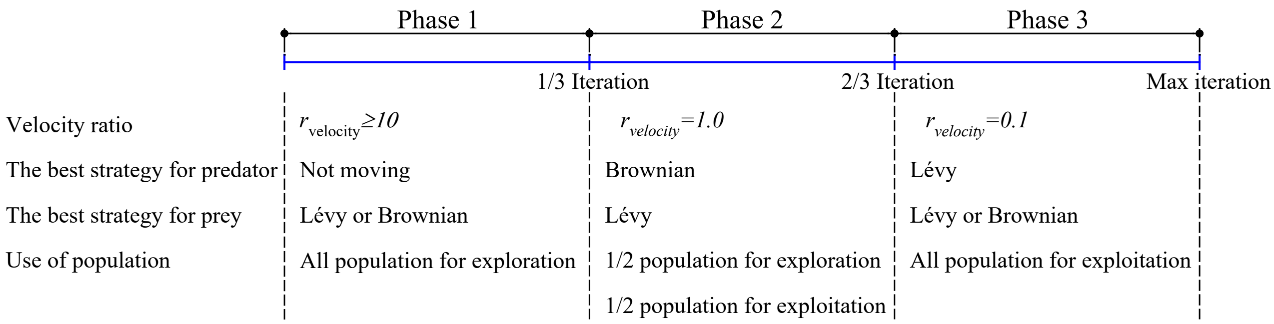



Fig. 1 Division of the number of iterations and the use of population in three main phases.

For an increase in the avoidance in local optima, efforts in looking for an abundant environment of the predator by suddenly taking a long, vertical jump are simulated in the algorithm. In the marine ecosystem, the formation of eddy or human activities related to fish aggregating devices (FADs) can change the behavior of the predator. These environmental issues can be considered as local solutions. Therefore, the sudden jump is an effort to escape from trapping in local minima. The environmental effects are calculated as:

$$\overrightarrow{\mathrm{Prey}} = \overrightarrow{\mathrm{Prey}} + CF \otimes \left[\vec{X}_{\min} + rand(p,u) \otimes \left(\vec{X}_{\max} - \vec{X}_{\min}\right)\right] \otimes \vec{U} \quad \text{if } rand \leq \text{FADs=0.2} \tag{21}$$

$$\overrightarrow{\mathrm{Prey}} = \overrightarrow{\mathrm{Prey}} + \left[FADs \times (1-rand) + rand\right] \times \left(\overrightarrow{\mathrm{Prey}}_{r1} - \overrightarrow{\mathrm{Prey}}_{r2}\right) \quad \text{if } rand > \text{FADs=0.2} \tag{22}$$

The binary vector $\vec{U}$ is built from the random vector *rand*, then each value in the vector $\vec{U}$ is compared with FADs=0.2. The return number equals 0 if the value is greater than 0.2, otherwise 1. The subscripts *r1* and *r2* are rows chosen randomly from the prey matrix. The symbol ⊗ means entry-wise multiplications. Two vectors $\vec{X}_{\max}, \vec{X}_{\min}$ are upper and lower bounds of updating parameters.

For more effective performance in convergence rate, the value of the fitness function should be saved and used in the next iteration. It represents the abilities of memory saving of the top predator or Elite. Therefore, in each iteration, the new fitness value is compared with the previous one. The current value is updated in the Elite matrix if its value is more suited. The step-by-step procedure of MPA are as follows:

1) Generate prey matrix, a random set of agents in search space, Equation (6)
2) Calculate the fitness value based on the obtained prey matrix, identify the top predator matrix using Equation (7), and replicate the top predator matrix to generate Elite matrix in Equation (8)
3) Implement the exploration and exploitation process based on 3 main phases:
   Phase 1: Update prey matrix using equation (10),
   Phase 2: Update prey matrix using equations (12) and (17),
   Phase 3: Update prey matrix using equation (20).
4) Evaluate fitness value based on newly obtained prey matrix, update Elite matrix, save the marine memory.
5) Simulate the effect of FADs, update the prey matrix using Equations (21) and (22).
6) Repeat steps (2) to (5) until the stopping criteria is met.

In this study, MPA is used in the model updating problem and FNN training. In the first application, the errors between an FE model of a steel beam and the corresponding experimental model are reduced throughout the optimization process of MPA. In the second application, MPA is utilized to improve FNN training as a stochastic trainer based on the optimization strategy. Three case studies are used to evaluate the effectiveness and feasibility of the training process. In the last step, the updated FE model is utilized to generate damage database for the training process. The trained networks are employed to predict damage localization and quantification. Implementation of the updated FE model and the trained networks in solving an SHM problem is presented in the next section.

### 2.3 How effective MPA improves Feedforward Neural networks training, MPAFNN

As discussed in section 2.2, heuristic optimization algorithms have been applied to enhance the learning process of FNNs. In this approach, the optimization algorithms can be used in several ways. Optimization algorithms can be used to identify a suitable architecture for an FNN. This means the number of hidden nodes, number of hidden layers, the proportion of dataset for training, validation, and testing are quantified. The second use of optimization algorithms is to adapt hyper-parameters e.g. momentum, learning rate for a good performance of a neural network. Another use is to identify a combination of connection weights and biases (threshold) in order to minimize errors of the neural network. The last is the objective of this study because connection weights and biases are extremely important variables in the training process. Therefore, weights $w_{ij}$ and biases $\theta_j$ are treated as particles or agents which are obtained from minimizing the fitness function in optimization algorithms. A set of weights and biases can be written as updating parameters in the optimization procedure, particles/agents = ($w_{ij}$, $w_{jk}$, $\theta_j$, $\theta_k$).

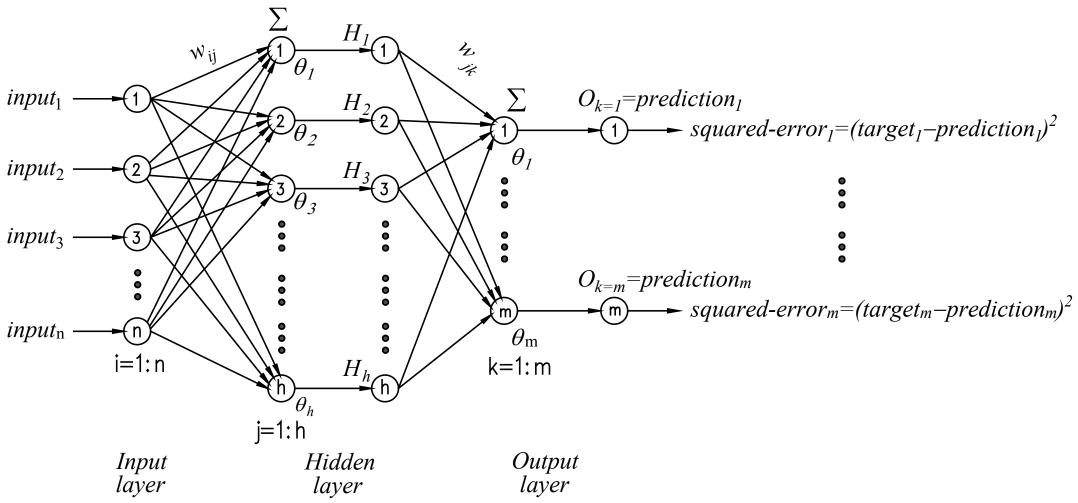



Fig. 2 Architecture of one hidden layer FNN

Objective function is another important matter in FNN training. Mean square error (MSE) is a popular metric and often is used as an objective function. The MSE with number of outputs *Nm*, and number of training samples *Ns* can be calculated as:

$$MSError = \frac{\sum_{i=1}^{Ns}\sum_{k=1}^{Nm}(\text{target}_{k(i)} - \text{prediction}_{k(i)})^2}{N_s} \tag{23}$$

where $\text{target}_{k(i)}$ and $\text{prediction}_{k(i)}$ denote respectively the real and predicted values of output $k^{th}$ when sample $i^{th}$ is used for training. Considering $w_{ij}$ and $w_{jk}$ be respectively connection weights from input node $i$ to hidden node $j$, and from hidden node $j$ to output node $k$ as in Fig. 2, the sum of the weights of inputs can be computed as:

$$sum_j = \left(\sum_{i=1}^{n} w_{ij} \times input_i + \theta_j\right), \text{ with } i=1 \textit{ to } n;\ j=1 \text{ to } h \tag{24}$$

where $\theta_j$ denotes the bias of hidden node $j$. Then, the outputs of the hidden nodes in the hidden layer by using the sigmoid activation function can be obtained as:

$$H_j = \sigma(sum_j) = \frac{1}{1+e^{-sum_j}} = \frac{1}{1+e^{-\left(\sum_{i=1}^{n} w_{ij} \times input_i + \theta_j\right)}}, \text{ with } i=1 \text{ to } 2;\ j=1 \text{ to } h \tag{25}$$

The weights sum of hidden layer and outputs of the output layer are determined as:

$$sum_k = \left(\sum_{i=1}^{n} w_{ij} \times H_j + \theta_k\right), \text{ with } k=1 \textit{ to } m;\ j=1 \text{ to } h \tag{26}$$

$$O_k = \sigma(sum_k) = \frac{1}{1+e^{-sum_k}} = \frac{1}{1+e^{-\left(\sum_{i=1}^{n} w_{ij} \times H_j + \theta_k\right)}} = prediction_k,\ k=1 \textit{ to } m;\ j=1 \text{ to } h \tag{27}$$

where $\theta_k$ denotes the bias of output node *k*, symbols *n, m, k* represent the number of input nodes, hidden nodes, and output nodes, respectively.

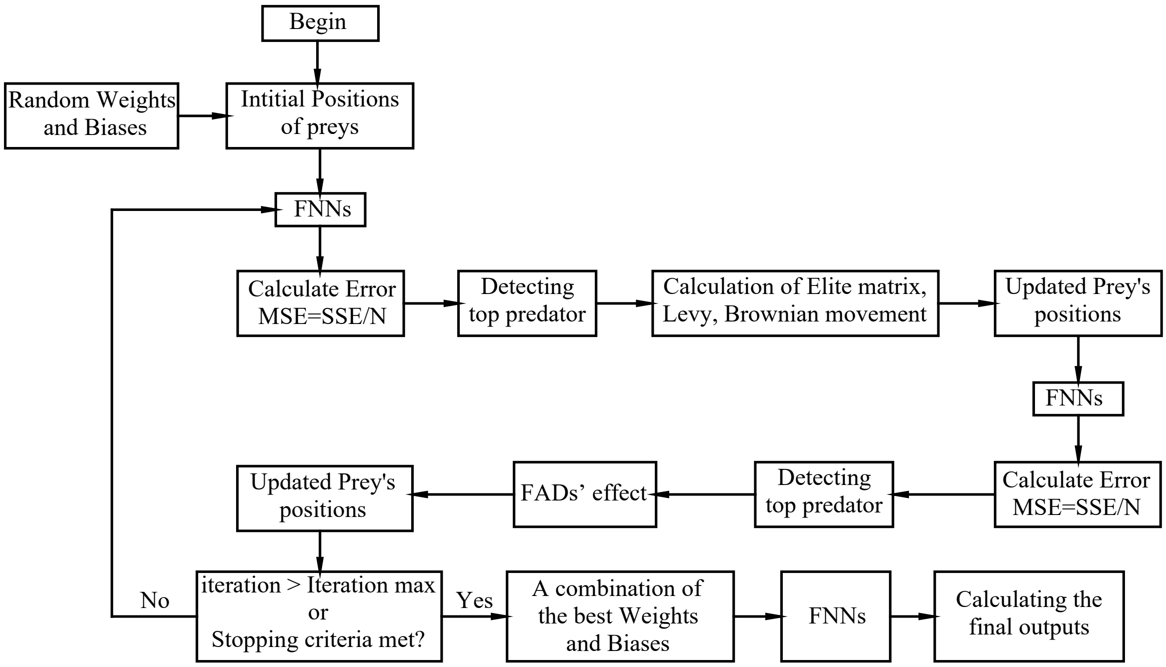



Fig. 3 The schematic training procedure for a feedforward neural network using MPA.

Based on the mentioned points, a procedure of training using a heuristic optimization algorithm, Marine predator algorithm (MPA) for FNN can be calculated as:

(1) Start with random values of initial positions of the preys, representing connection weights and biases.
(2) Training samples are treated as inputs and outputs. These values are normalized in a range of 0.01 to 0.99. Large inputs can cause a very flat transfer function or a saturate neural network. Therefore, learning ability can be reduced. Large outputs are not necessary because the sigmoid function produces only a maximum absolute value approximately up to 1.0. A very small value also affects the accuracy of the computation. An interval [0, 1] is the common value that is returned by the sigmoid function. The value is only asymptotic to 0 or 1.
(3) Calculate the mean square error (MSE) of all training samples and use the obtained MSE as the fitness in MPA. The optimization process of MPA is started and evolves the initial weights and biases to reduce the MSE. A new set of connection weights and biases is achieved at the end of the optimization process.
(4) The training process continues until the termination criterion is met, i.e. the maximum iteration. The best values of weights and biases are used to approach the highest classification in case study 1 and the highest accuracy of prediction in two case studies 2 and 3. The progression of using MPA in FNN training is shown in Fig. 3.

## III. Case studies:

### *3.1 Case study 1: Classification benchmark problems*

In this section, the effectiveness of the proposed approach in improving feedforward neural network training is investigated through some classification benchmark problems. Some well-known algorithms such as particle swarm optimization (PSO), gravitational search algorithm (GSA), hybrid algorithm PSOGSA, and grey wolf optimizer (GWO) are used for comparison. PSO, GSA are famous optimization algorithms in the scientific community. The effectiveness of the two algorithms has been confirmed through vast studies. PSOGSA is an improved version of PSO and GSA. It combined strong points of both considered algorithms. A novel GWO confirms that it could obtain very competitive results compared to prominent algorithms, especially in engineering problems. Application in the engineering field is the main aim of this study. Therefore, the reliability and efficiency of the proposed approach MPAFNN can be comprehensively evaluated by comparative studies using these approved algorithms. The average of MSE, the best MSE, the standard deviation on Equation (28), convergence curves and classification rate are comparative objectives. The classification rate is calculated based on the best MSE. Each algorithm is run independently 20 times in order to calculate average values as well as standard deviations of MSE.

$$stdMSE = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(MSE_i - \overline{MSE}\right)^2} \quad (28)$$

Where $N$=20 is the number of run, $MSE_i$ represents the observed values of vector $MSE$ after $N$ runs, and $\overline{MSE}$ indicates the mean value of these observations.

Table 1 Classification datasets

| Problem | Number of input nodes | Number of hidden nodes[1] | Number of classes | Number of training / testing | Runs |
|---|---|---|---|---|---|
| Balloon | 4 | 4×2+1=9 | 2 | 16 / 16 | 20 |
| Iris | 4 | 4×2+1=9 | 3 | 150 / 150 | 20 |
| Breast cancer | 9 | 9×2+1=19 | 2 | 599 / 599 | 20 |
| Heart | 22 | 22×2+1=45 | 2 | 80 / 80 | 20 |

Four investigated benchmarks are balloon, Iris, breast cancer, and heart dataset (source: http://www.ics.uci.edu/~mlearn/MLRepo sitory.html). Specifications of all databases are described in Table 1. For a stable performance, each classification problem is solved in 20 independent runs. The collected results from these 20 runs are utilized for comparison.

The initial parameters for each problem are set as follows:

- As for balloon problem the population size $p$=50, maximum iteration $iter_{max}$=250 because of its training data is more simple than that of the others.
- Regarding the Iris, breast cancer, and heart problems, the population size $p$=200 and maximum iteration $iter_{max}$=250.
- GSA: coefficient $\alpha$=10, initial value $G_0$=2
- PSO: cognitive coefficient $c_c$=2, social coefficient $c_s$=2, inertial coefficient $w_{min}$=0.5, $w_{max}$=0.9, $w = w_{\min} - iter_{current} \times (w_{\max} - w_{\min}) / Iter_{\max}$
- PSOGSA: $c_c$=1, $c_s$=1, $\alpha$=20, $G_0$=1, $w = w_{\min} - iter_{current} \times (w_{\max} - w_{\min}) / Iter_{\max}$ with $w_{min}$=0.5, $w_{max}$=0.9.
- GWO: vector $\vec{a} = 2 \times (1 - iter_{current}) / Iter_{\max}$ , linearly decreases from 2 to 0.
- MPA: $FADs$=0.2, $P$=0.5, $\alpha$=1.5
- Architectures of FNN for balloon, Iris, breast cancer, and heart are 4-9-1, 4-9,1, 9-19-1, 22-45-1 respectively. During the training process, using random orders of the given dataset in each loop.
- List of stochastic trainers: GSAFNN, PSOFNN, PSOGSAFNN [32-33], GWOFNN [34], and proposed MPAFNN

[1] Note that this study doesn't focus on identifying the number of hidden nodes. The number of hidden nodes is suggested to be calculated as: (number of input nodes) × 2 + 1

*3.1.1 Benchmark 1: Balloon*

From Table 1, the total number of dimensions (or updating parameters), i.e. connection weights and biases, is 55. It is the simplest dataset among the others in this study (see Table 1). Therefore, it is understandable when the classification rate of algorithms exceed GSA's, i.e. 90% (see Table 2). It can be seen that MPAFNN has an outstanding performance compared with the other algorithms for all criteria e.g. average, standard deviation, and the best MSE. The best average MSE, as well as std MSE, indicate that MPAFNN owns the highest capability in local optima avoidance. This shows a good potential of MPAFNN in the training process.

Table 2 Classification results for balloon dataset

| Values | GSAFNN | PSOFNN | PSOGSAFNN | GWOFNN | MPAFNN |
|---|---|---|---|---|---|
| Average MSE | 1.92E-01 | 1.69E-01 | 9.58E-07 | 6.08E-07 | **4.21E-09** |
| The best MSE | 4.83E-02 | 1.00E-04 | 3.54E-08 | 6.85E-09 | **3.06E-16** |
| std MSE | 7.94E-02 | 6.71E-02 | 2.13E-06 | 2.09E-06 | **8.21E-09** |
| Classification rate (%) | 90 | **100** | **100** | **100** | **100** |

*3.1.2 Benchmark 2: Iris*

Table 3 Classification results for Iris dataset

| Values | GSAFNN | PSOFNN | PSOGSAFNN | GWOFNN | MPAFNN |
|---|---|---|---|---|---|
| Average MSE | 9.23E-02 | 7.94E-02 | 4.63E-03 | 4.99E-03 | **2.24E-03** |
| The best MSE | 4.00E-02 | 3.90E-02 | 3.40E-03 | 3.29E-03 | **1.58E-03** |
| std MSE | 6.23E-02 | 3.09E-02 | 1.10E-03 | 1.99E-03 | **5.05E-04** |
| Classification rate (%) | 33.33 | 72.67 | 98 | 98.67 | **99.33** |

Iris flower classification is a popular problem and often is used as a benchmark example in network training. The number of variables to be optimized in Iris problem is the same as the balloon. The larger number of training samples, i.e. 150, affects the accuracy of the prediction regarding all the considered algorithms for solving this problem. There is a slight decrease in the classification rate of PSOGSAFNN, GWOFNN, and MPAFNN as can be seen from Table 4. However, the prediction ability of the proposed algorithm is very impressive with a classification rate of over 98%. As such, MPAFNN continues to outperform the other algorithms regarding all comparative criteria.

*3.1.3 Benchmark 3: Breast cancer*

It is the most difficult dataset that is used for training in this study with 210 updating parameters and 599 training samples. The best MSE and standard deviation MSE belong to PSOGSAFNN and GWOFNN, respectively. MPAFNN shows its second-best position related to both indexes. Besides, MPAFNN proves that it is the best algorithm for average MSE over 20 runs and classification rate at $1.79\times10^{-2}$ and 98% in Table 4, respectively. The complication in the dataset like a large number of variables and training samples makes the obtained achievements of the proposed algorithm more meaningful.

Table 4 Classification results for breast cancer dataset

| Values | GSAFNN | PSOFNN | PSOGSAFNN | GWOFNN | MPAFNN |
|---|---|---|---|---|---|
| Average MSE | 1.41E-01 | 1.28E-01 | 1.89E-02 | 1.91E-02 | **1.79E-02** |
| The best MSE | 6.02E-02 | 7.02E-02 | **1.44E-02** | 1.63E-02 | 1.46E-02 |
| std MSE | 8.21E-02 | 3.10E-02 | 2.36E-03 | **9.41E-04** | 1.51E-03 |
| Classification rate (%) | 57.51 | 92.85 | 96.42 | 96.85 | **98** |

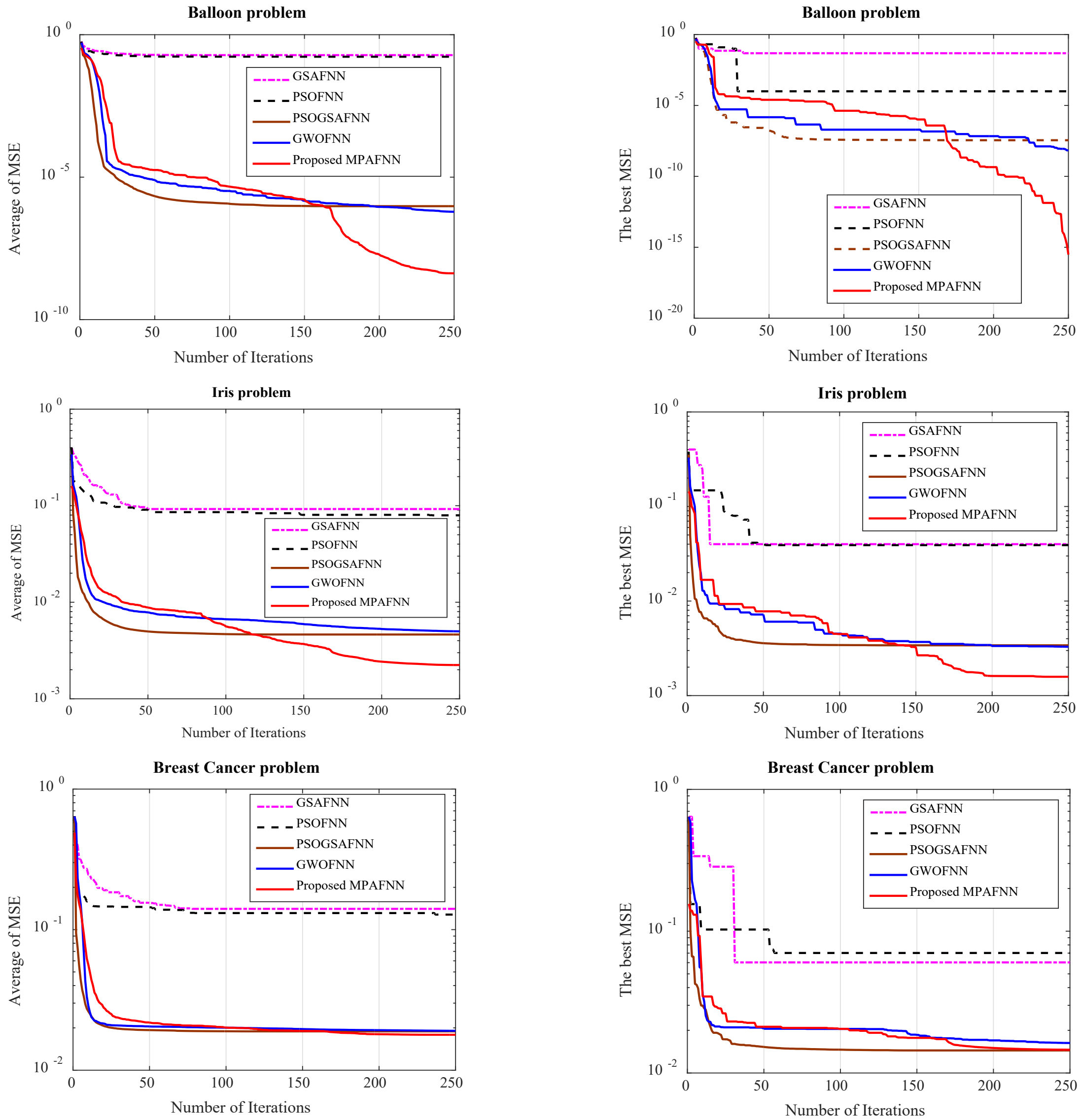
*3.1.4 Benchmark 4: Heart disease*

This classification problem has the largest number of variables to update, 1081. It is also the most complex model for FNN compared with the others. However, the proposed approach MPAFNN solves this problem with superior results. Table 5 reveals that MPAFNN occupies the best position for all indicators. The obtained results continue to confirm the potential in local avoidance and high accuracy of the proposed algorithm, i.e. MPAFNN, in training a neural network.

Table 5 Classification results for heart dataset

| Values | GSAFNN | PSOFNN | PSOGSAFNN | GWOFNN | MPAFNN |
|---|---|---|---|---|---|
| Average MSE | 3.23E-01 | 2.21E-01 | 1.07E-01 | 8.22E-02 | **3.05E-05** |
| The best MSE | 2.61E-01 | 1.94E-01 | 6.76E-02 | 6.68E-02 | **3.58E-18** |
| std MSE | 4.37E-02 | 3.43E-02 | 2.34E-02 | 9.36E-03 | **1.63E-05** |
| Classification rate (%) | 50 | 71.25 | 92.5 | 92.5 | **100** |

Apart from the statistical results as discussed above, MPAFNN also obtains the best convergence rate using both average and the best MSE for the balloon, Iris, and especially heart disease problems. For the most complex dataset, i,e, breast cancer, MPAFNN still shows superior performance compared with PSOGSAFNN, GWOFNN and much better results than GSAFNN, PSOFNN as depicted in Fig. 4.



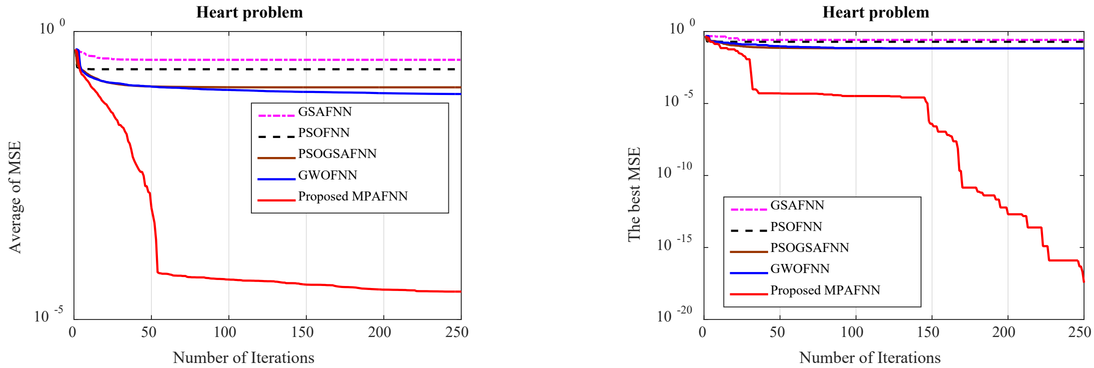



Fig. 4 Convergence curves for FNNs based on averages and the best MSE over N=20 runs.

From four benchmark problems, the obtained results associated with MSE and convergence study proves that MPAFNN is the most suitable algorithm for training neural network due to its high stability, precision and accuracy.

### *3.2 Case study 2: Damage detection in structures using numerical data*

In this section, two numerical simulated cases are used to investigate the capability of the proposed hybrid approach in damage detection of beam-like structures. As mentioned in section 2.1, flexibility-based curvature indices along the considered structures are used to identify damage location and corresponding severity. Each set of flexibility-based curvature indices is calculated by using modal properties e.g. frequencies and mode shapes from two intact and damaged states of the structures. Therefore, each stiffness reduction at a position or several positions of the investigated beam generates a set of flexibility-based curvature indices along the beams. The inputs are these damage indices while the target outputs are damage location and severity. The collected dataset then is divided into 80% - 20% for training and test, respectively.

#### *3.2.1 Training procedure*

The proposed algorithm is used to train neural networks for fault assessment in structures. The applicability of the algorithm to engineering problems was evaluated first by a numerical study before applying it to a real structure. The training process begins by acquiring the values of flexibility-based curvature at all measured points along with two considered structures. These indicators are treated as inputs of the FNN. Due to the supervised learning process, each input dataset has its corresponding target outputs. These target outputs consist of location and severity of damage that are used initially to simulate the inputs. Depending on each particular problem, the number of output nodes, $m_{nodes}$ is different. For instance, for localization and quantification of single damage, the number of output nodes is $m_{nodes}$=2, one for location, and the other for severity. In the two damage cases of this study, with the assumption of the same failure levels for all damage, the number of output nodes is $m_{nodes}$=3, two for location, and the rest for severity. Choosing the number of hidden nodes is an important part of the FNN structure. A small number of hidden nodes speeds up convergence process meanwhile a larger number of hidden nodes increases the accuracy of predicted results. In this study, the optimal number of hidden nodes is not the main objective. Therefore, the number of hidden nodes is calculated by $S_{nodes} = n_{nodes} \times 2 + 15$, where $S_{nodes}$ and $n_{nodes}$ denote respectively the number of hidden nodes and input nodes. An architecture of FNN with $n_{nodes} - n_{nodes} \times 2 + 15 - m_{nodes}$ is used in this study for damage detection. The training process of FNN is depicted in Fig. 5.

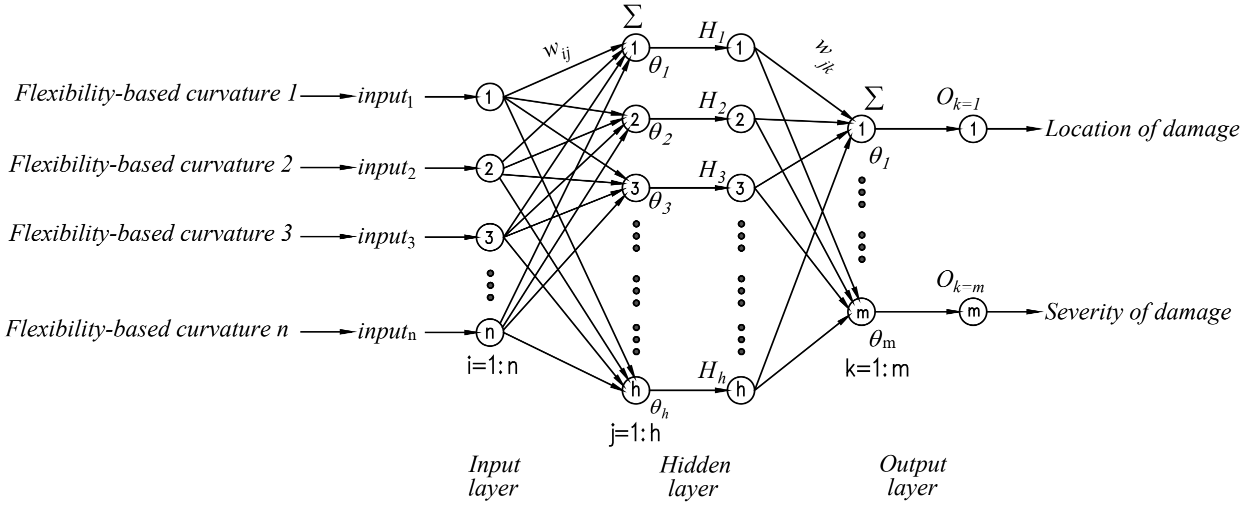



Fig. 5 Architecture of FNN, n-S-m for the tested beam.

*3.2.2 Single damage in a two-span continuous beam*

*a. Training dataset:*

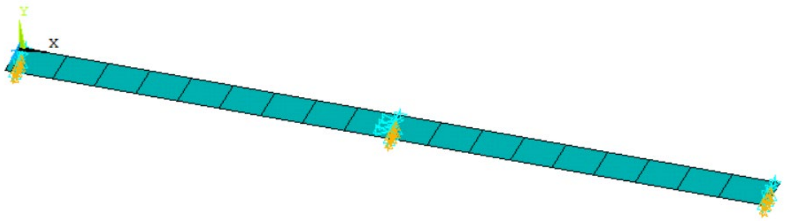


Fig. 6 FE model of the two-span continuous beam with 20 elements.

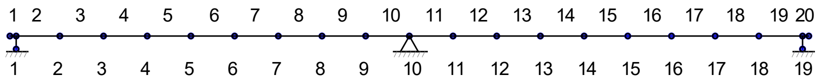
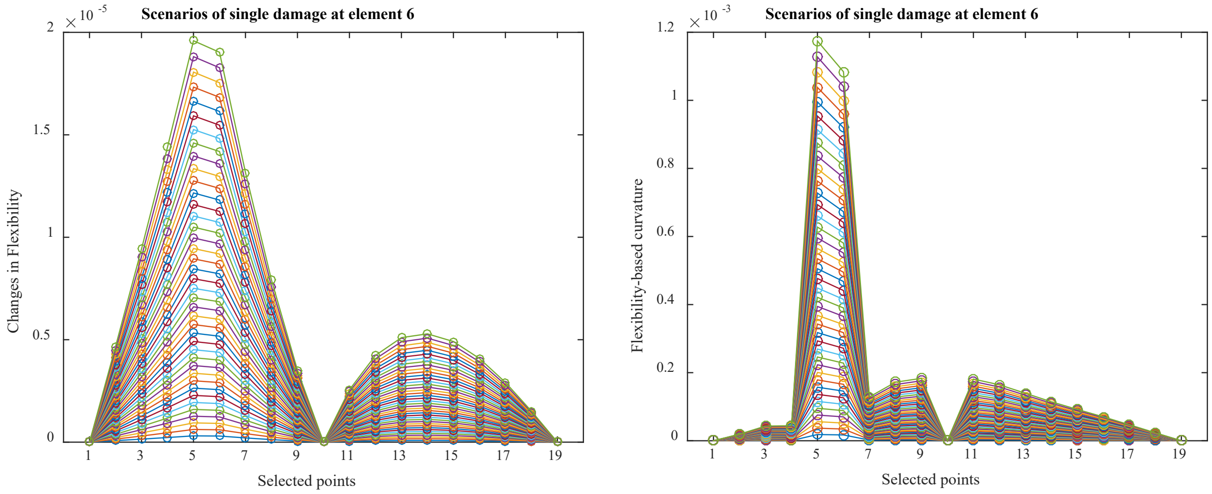



Fig. 7 Schematic division of elements, nodes in the beam and boundary conditions.

A two-span continuous steel beam with a length of 1.28 m, and a rectangle cross-section of 0.07x0.01 (m) is modelled in ANSYS using SHELL181 elements (see Fig. 6). Details of this element are introduced in the next section 4.2.2. Material properties used in the simulation are Young's modulus $E$=2×10$^{11}$ N/mm$^2$, density $\gamma$=7820 kg/m$^3$, and Poisson's ratio ν=0.3. Boundary conditions comprise one fixed support at the middle and two movable supports at two ends (node 1 and node 19) as in Fig. 7.

Data from single damage scenarios was created for training based on the assumption of the stiffness decrease (by reducing Young's modulus) of each defective element. The stiffness of each element was reduced from 1% to 40% with an interval of 1%. Therefore, 18 elements × 40 cases = 720 scenarios were used to generate modal properties of the damaged beam. The first five natural frequency values and displacement mode shapes at nineteen nodes, as labelled in Fig. 7, were collected to calculate the flexibility-based curvature indices. Fig. 8 depictures the flexibility-based damage indices using Equations (4) and (5) due to damage at element 6 on the beam. It can be seen that the use of curvature for monitoring changes in flexibility (calculated by Equation (5)) between intact and damaged states can provide a clearer view of the damaged element (see Fig. 8b).




a. Using flexibility changes b. Using flexibility-based curvature

Fig. 8 Comparison of the flexibility-based damage indices associated with the stiffness reductions from 1 to 40 %.

A feedforward neural network (FNN) with a structure 19-53-2, 80% of 720 = 576 samples and 1168 variables e.g. connection weights and biases, are used to localize and quantify damage in the continuous beam. To ensure the accuracy of predicted results, some initial parameters of marine predator algorithm (MPA) are preset: FADs=0.2, $P$=0.5, $\alpha$=1.5, population = 400, maximum iteration = 15,000.

*b. Testing dataset*

The testing dataset includes two sets. The first set was derived from the collected samples by 20% of 720, i.e. 144 samples. The second one was based on three extra single damage scenarios generated by reducing the stiffness with 15.5%, 22.5%, 26.5% at elements 5, 8, 9, respectively. The former is used to evaluate the regression ability of the proposed approach, while the latter is used to perform visual results of damage prediction using MPAFNN. Modal properties of these scenarios were collected to calculate the flexibility indices. The trained neural network was used to predict damage location and corresponding severity based on these obtained flexibility indices.

*c. Results:*

After 15,000 iterations, the convergence curve for FNN is plotted in Fig. 9. It can be seen that the convergence curve includes three major curves related to 3 main phases in the optimization process. The first curve with iteration ≤1/3 of $Iter_{max}$, i.e. 5000, shows stepwise behaviour meanwhile the last curve with iteration from 2/3 of $Iter_{max}$ to $Iter_{max}$, i.e. from 10,000 to 15,000, is smoother. It is possible to understand that the first phase only focuses on exploration. All agents try to forage the global optimum point in the overall search space. The search results are improved slowly during the passage of iterations. The main task of the last phase is exploitation. Therefore, when iterations pass, the predators and preys try to refine the solutions as much as possible. In phase 2, a half population continues to explore better solutions while the other begins to refine the solutions. Nevertheless, its curve is smoother than that of phase 1, and unsmooth compared with phase 3 (see Fig. 9)

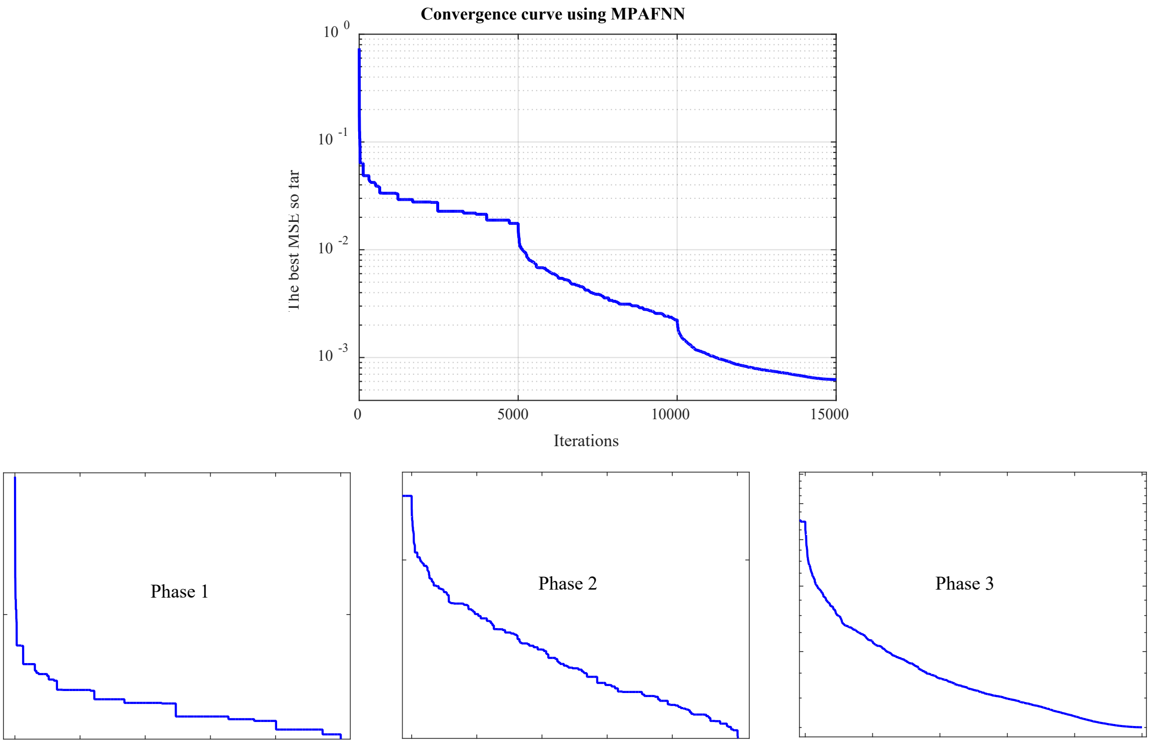
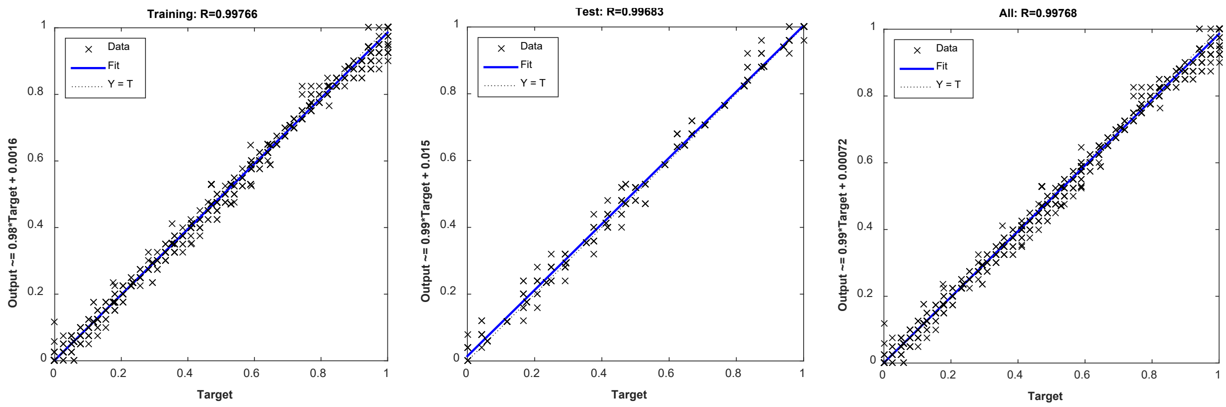



Fig. 9 Convergence curves for training process based on the best MSE of objective function.

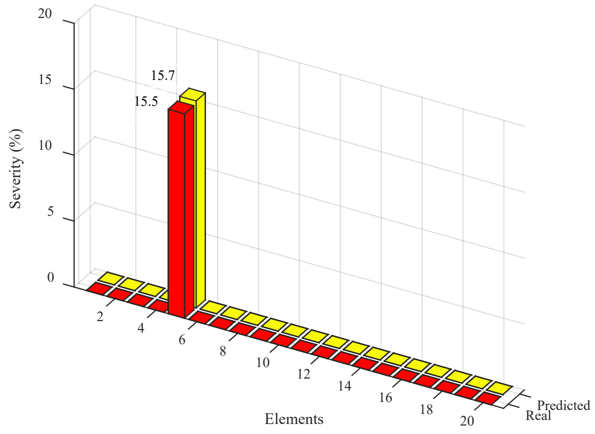
The set of connection weights and biases obtained from the training process is used to predict damage scenarios based on the testing datasets. The targets and the corresponding predicted outputs for training, test and all sets are used to build regression plots. In general, network performance is assessed based on the angle of the fit line in the regression plot as well as R values. From Fig. 10, the data almost locates along a 45-degree line. This confirms the accuracy of the predicted values by MPAFNN. Besides, high R values (over 0.99) are clear proof of a good agreement for all data sets.



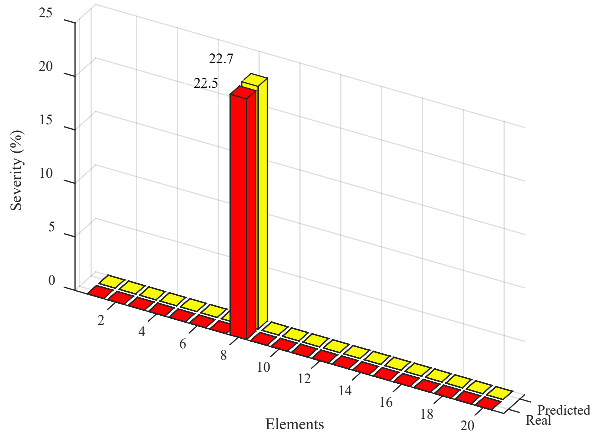

a. Training regression b. Test regression c. Overall regression

Fig. 10 The regression graphs using MPAFNN for the two-span continuous beam

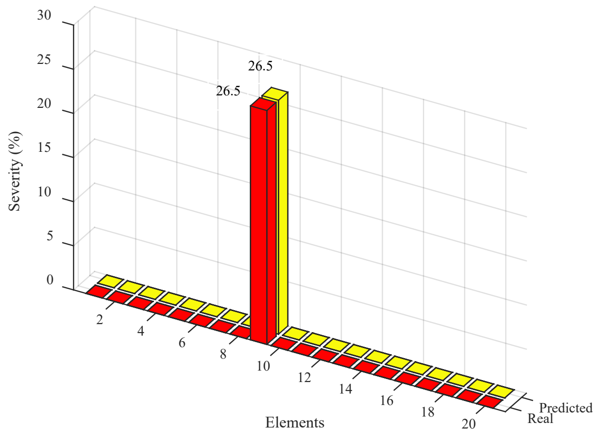
Results of the three extra damage scenarios are plotted in Fig. 11. The proposed algorithm localizes the exact damage position for all scenarios. The errors associated with these predictions are 0.2%, 0.2%, and 0% at element 5, 8, and 9 respectively. For the first case, therefore, the proposed approach shows its potential for real applications.




Damaged element: 5th




Damaged element: 8th




Damaged element: 9th

Fig. 11 Single damage scenarios

*3.2.3 Different damage scenarios in a simply supported beam problem.*

*a. Training dataset:*

In this section, SHELL181 elements are employed to model a one-meter simply supported steel beam (see Fig. 12). The dimension of cross-section and material properties are the same as the continuous beam of the previous section. The structure has two supports, one movable support and one fixed support at nodes 1 and 15 respectively. Details of the element number, nodes and boundary conditions are shown in Fig. 13.

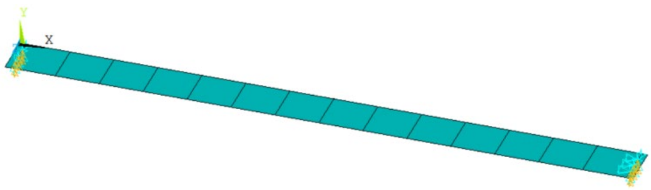


Fig. 12 FE model of the simply supported beam with 16 elements

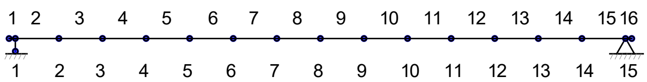
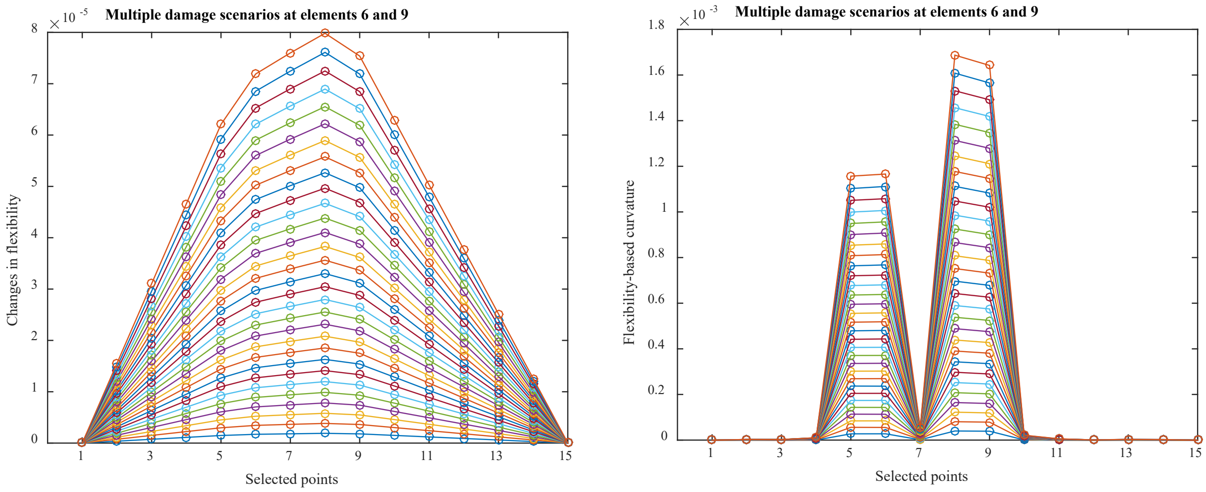



Fig. 13 Schematic division of elements, nodes on the beam and boundary conditions.

In each damage scenario, the stiffness of two elements is reduced by a factor in an interval from 1 to 30% with an interval of 1%. The aim of this step is to evaluate the feasibility of the proposed approach for multiple damage detection regarding the considered structure. Therefore, to facilitate collecting data and reducing the computational time, only 7 successive elements from 5th to 11th are used to generate data from the damaged beam. Number of damage scenarios $N_{data}$ can be, therefore, calculated as the number of damage levels multiplied by the number of combination of 7 elements chosen 2 at a time, i.e. $N_{data} = 30 \times C_7^2 = 30 \times 21 = 630$.

Thirty damage scenarios with defective elements 6 and 9 are used to plot the change of curvature in the modal flexibility (Fig. 14b). Although two damage locations can be revealed at elements 6 and 9 based on sudden slope shifts in the curve of flexibility changes in Fig. 14a, the localizations of defects are much better in Fig. 14b.




a. Using flexibility changes b. Using flexibility-based curvature

Fig. 14 Comparison of the flexibility-based damage indices associated with the stiffness reductions from 1 to 30 %

Both damages are identified by using an architecture 15-45-3, 80% of 630= 504 samples and 858 variables. The same values of the initial parameters for MPAFNN in the first numerical study are employed to solve this case.

*b. Testing the dataset*

Likewise to section 3.2.2.b, two datasets were generated for test 20% of 630, i.e. 126 samples were used for regression capacity. For visual results, four multiple damage scenarios are considered regarding 4 pairs of elements 7&10, 8&9, 5&9, 8&10 with damage severities 15.5%, 20.5%, 9.5%, and 12.5%, respectively. The first five frequencies and displacement mode shapes at fifteen equidistant points were then used as testing data.

*c. Results:*

Fig. 15 shows a similar trend of convergence rate in three phases as discussed in section 3.2.2.c. Three separate curves represent three phases for the exploration and exploitation of MPA. The refinement of the best solution continues to be present in the last phase by a smoother curve compared with the others.

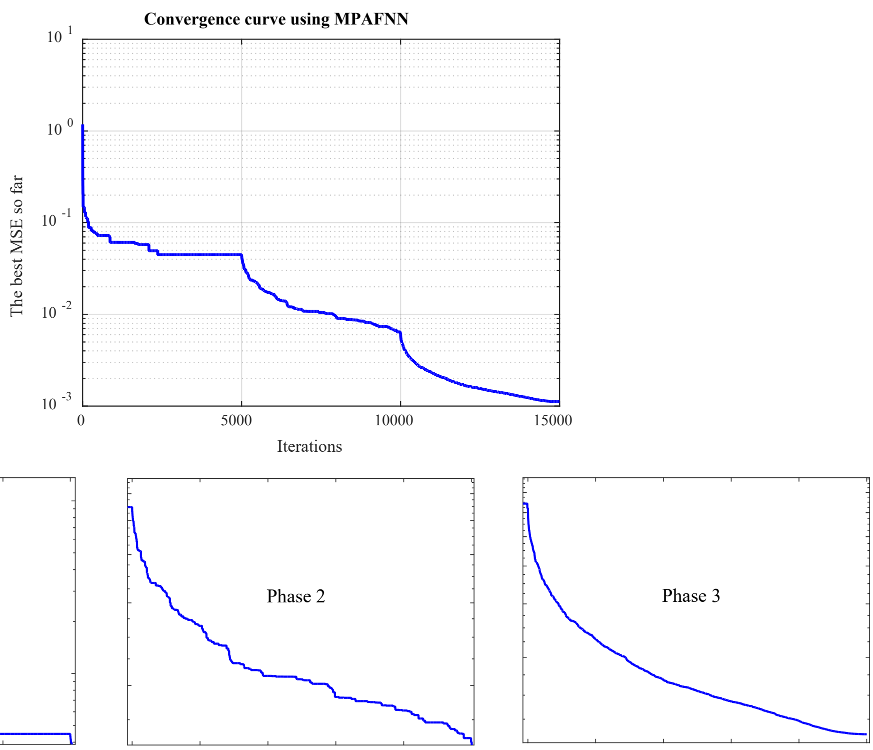
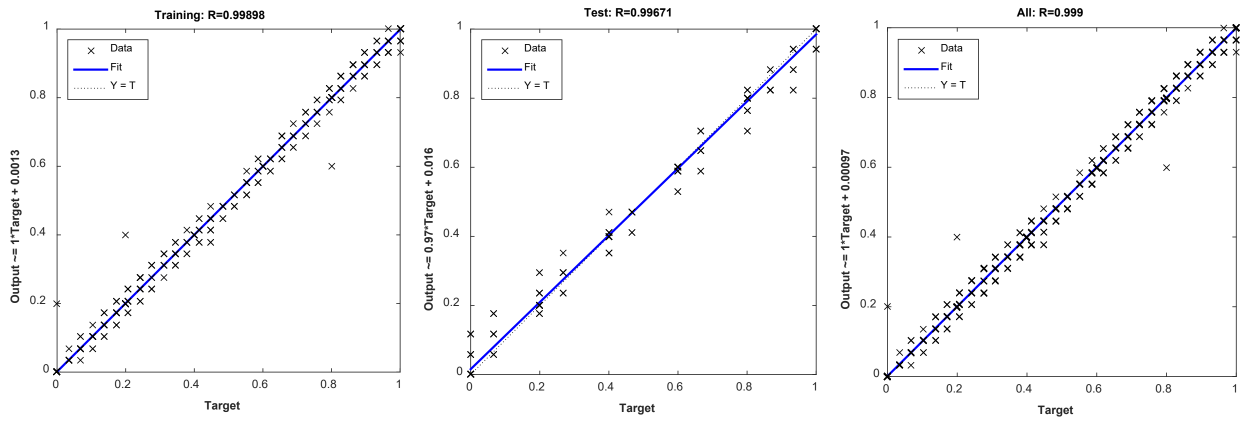



Fig. 15 Convergence curves for training process based on the best MSE of the objective function.

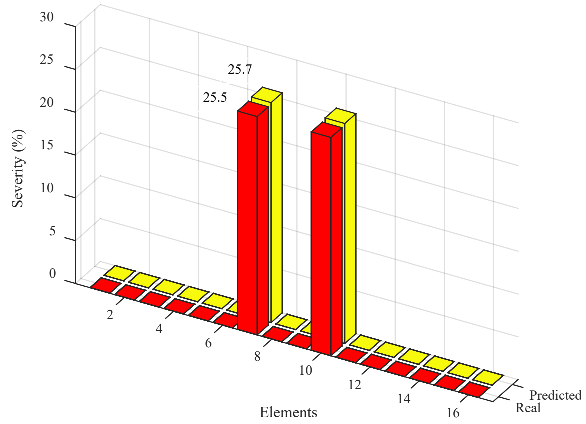
The task of MPAFNN in the second scenario is more difficult than the first one. The proposed algorithm has to localize two positions and the corresponding extent of the damage. The predicted values of 126 test samples were compared with the given ones. Superior fits continue to be shown via regression plots for training, test and all data as in Fig. 16. It can be seen that a 45-degree line is obtained and all R values are larger than 0.997.



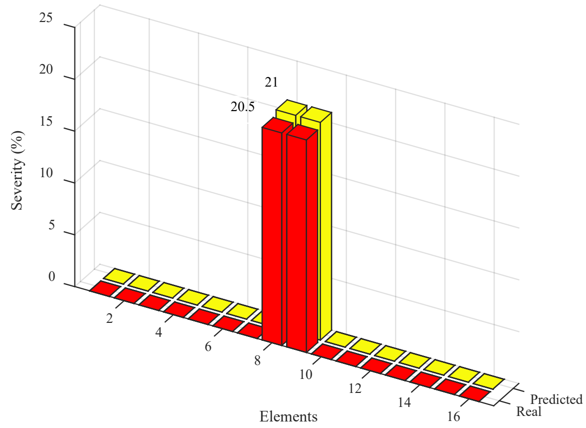

a. Training regression b. Test regression c. Overall regression

Fig. 16 The regression graphs using MPAFNN for the simply supported beam

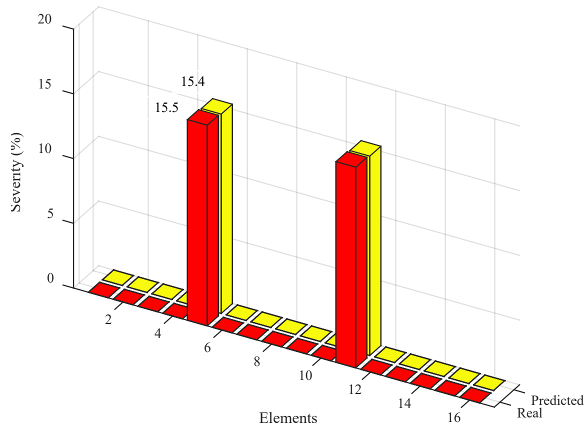
Predicted results successfully identify damage at 7&10, 8&9, 5&11, 8&10 in complete agreement with real damaged elements (see Fig. 17). Differences in damage severity between target and prediction are 0.2%, 0.5%, -0.1%, and 0.3% as in Fig. 17a, b, c, and d respectively.



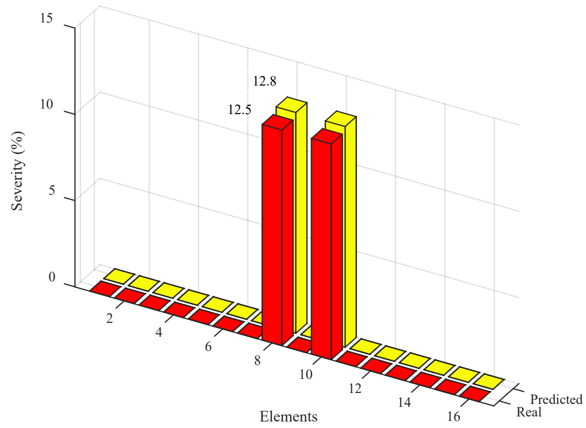

a. Damages at elements: 7 & 10




b. Damages at elements: 8 & 9




c. Damages at elements: 5 & 11




d. Damages at elements: 8 & 10

Fig. 17 Identification of multiple damages

From obtained results in the numerical study, it can be said that the application of MPAFNN in damage identification is feasible. For this reason, MPAFNN is used to identify damage in a real structure using experimental vibration data. Flexibility-based curvature index continues to be used as input for FNN.

### *3.3 Case study 3: Damage detection in a laboratory beam with noise effect:*

Another important goal of this study is to use the marine predator algorithm for model updating based on experimental modal properties of a laboratory beam. Then the proposed hybrid method is used to identify structural damage of the updated FE model under different noise levels. Likewise to the previous section, differences between flexibility-based curvature of the healthy and damaged states are employed to identify damage location as well as severity. However, training of the neural network is only efficient when the dataset for training is large enough and well distributed. Therefore, a measurement campaign on an intact laboratory beam was carried out to determine the modal characteristics. A baseline finite element model (FEM) of the tested intact beam is built using the experimental data. The FE model updating process is a crucial problem that can determine the quality of the updated model, especially when it is used for the detection of structural damage. The better agreement between the experimental and simulated model, the more accuracy of data training for damage prediction is guaranteed. Collected data from the baseline model serves as training data. The white Gaussian noise level of 2% was added to the modal properties e.g. frequencies and mode shapes in intact and damaged states to consider noise effect on data training.

#### *3.3.1 Steel beam measurement campaign:*

*a. Experiment description:*

The studied example is a one-meter intact steel plate with a rectangular cross-section dimension of 70×10 (mm$^2$). Unidentified forces induced by an impact hammer were used to generate a free vibration of the intact beam in the laboratory. Under this excitation, this kind of tested structure will create high amplitude vibration levels. Therefore, sensors with lower mass ($m_{sensor}$=5.8g) and sensitivity are chosen. In this experiment, the sensitivity range of accelerometers was in an interval (10.13 – 10.50) mV/m·s$^{-2}$. This guarantees that the frequency range of interest from 0.5 to 10,000 Hz can be obtained and clipping of the response is avoided when vibration is not inside the accelerometers' range. Besides, the lower weight of the sensor can also generate insignificant effects on the vibrational properties of the tested structure. To facilitate installing and removing accelerometers, accessories such as mounting bases or adapters were used to mount the sensors to the beam surface. The sensors were stud mounted to the bases or adapters after these accessories were directly glued to the beam surface. Fifteen accelerometers were placed at 15 equidistant points along the beam to obtain dynamic responses. By means of numerical studies, the mode shape obtained from 15 equidistant points is smooth enough for damage detection when the first five modes are used. The effects of sensor weight and wire on the vibration of the beam were considered in the FE model. A steel frame was manufactured to hang the tested beam on two 0.8 mm steel wire at two points 4 and 12. Since the experiment was conducted in the laboratory, the effects of environmental factors, such as humidity and temperature, on measured data can be significantly reduced. Schematic sensor placements including several sensors and the location of each sensor for the intact beam are shown in Fig. 18. One striking impact was used and data acquisition time was around 5 minutes at a sampling rate of 2651Hz. Data acquisition system, placement of accelerometers in the laboratory are shown in Fig. 19.

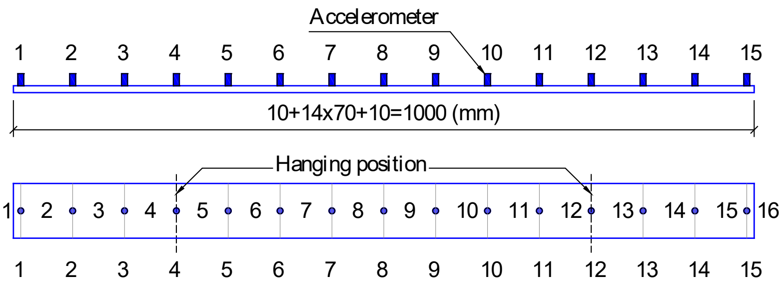
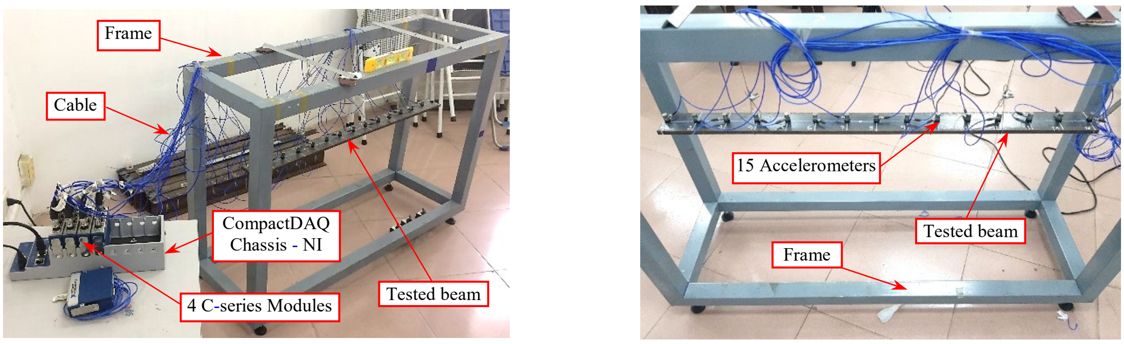



Fig. 18 Sensor placement at 15 equidistant points and element division of the intact beam




a. Data acquisition system (DAS) b. Steel frame and tested beam

Fig. 19 Overview of sensor placement on the beam surface and data acquisition system

*b. Data processing and results of system identification:*

In this section, the experimentally measured data are processed to determine modal parameters. For the output-only modal analysis, covariance-driven stochastic subspace identification algorithm (SSI-COV) is used, showing clear advantages in computational time as well as accuracy [35-36]. A stabilization diagram was used to distinguish spurious and physical modes. Some strict stabilization criteria are pre-set as follows [37-38]:

$$(f_p - f_{p+1}) \times 100 / f_p < 1\% \quad \text{for frequency} \tag{29}$$

$$(1 - MAC_{p,p+1}) < 1\% \quad \text{for modal vector} \tag{30}$$

$$(\xi_p - \xi_{p+1}) \times 100 / \xi_p < 5\% \quad \text{for damping factor} \tag{31}$$

where, MAC is modal assurance criterion, used to perform the correlation between two modal vectors $p$ and $p$+1. Vertical lines in the stabilization diagram are plotted in an interval (0 – 800) Hz. For a clear vision, the frequency response of the beam is also displayed in the stabilization diagram. Five peaks in frequency response reveal obtained modes from system identification. The first five natural frequencies and the corresponding mode shapes of the intact beam are shown in Fig. 21. Natural frequencies of both damaged and undamaged beams are shown in Table 6.

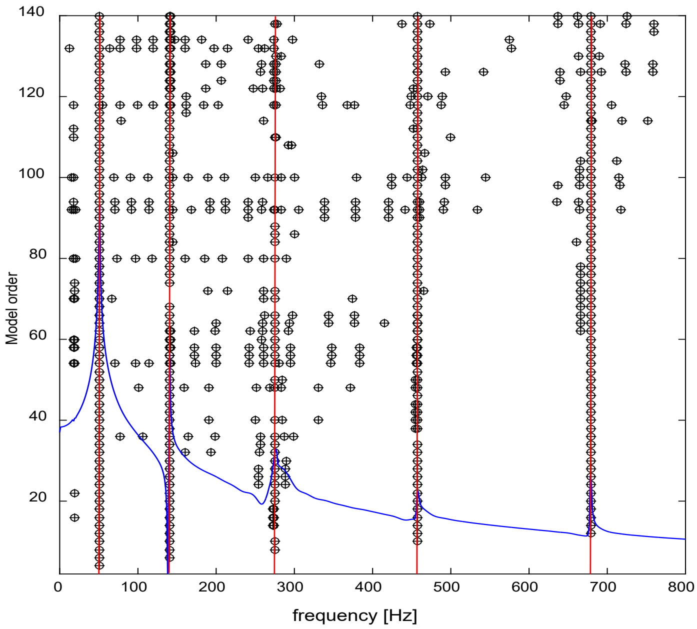



Fig. 20 Stabilization diagram.

Table 6 Summary of natural frequencies for the healthy beam

| Mode | Frequency | Type of mode |
|---|---|---|
| 1 | 50.83 | 1st vertical bending mode |
| 2 | 140.40 | 2nd vertical bending mode |
| 3 | 274.74 | 3rd vertical bending mode |
| 4 | 456.94 | 4th vertical bending mode |
| 5 | 678.90 | 5th vertical bending mode |

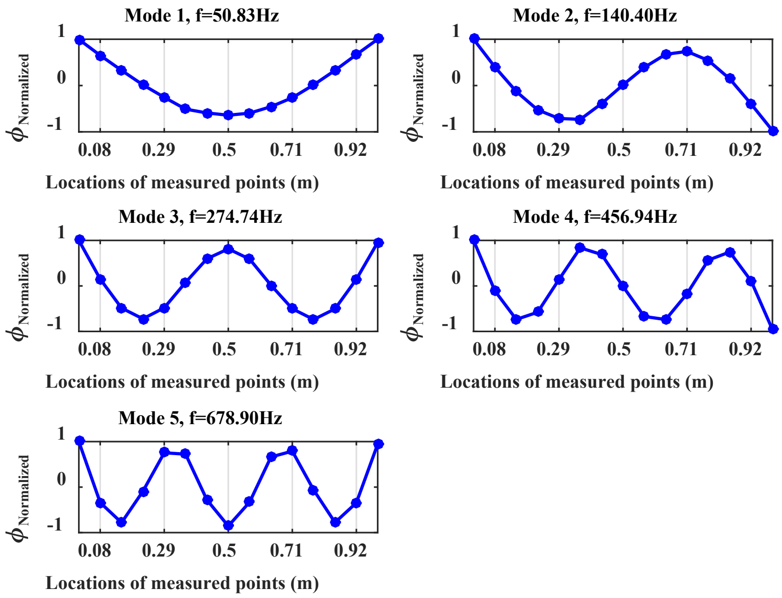
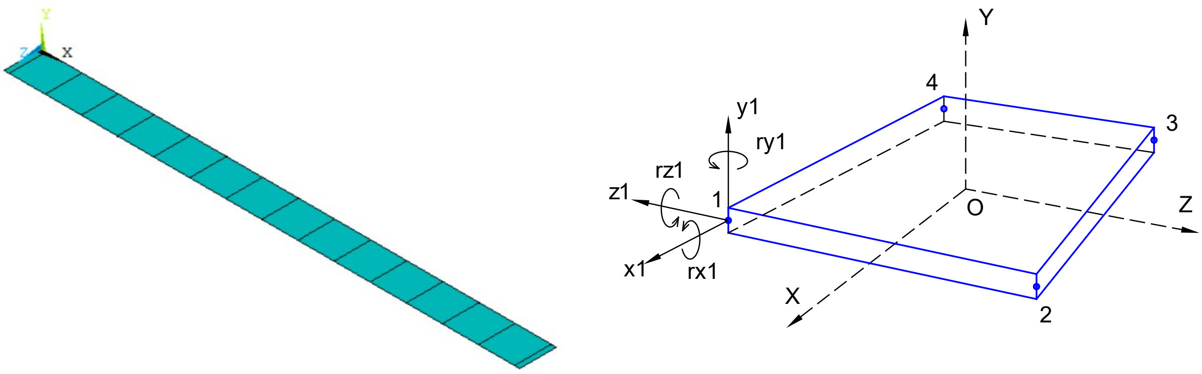



Fig. 21 The first five vertical bending modes of the intact beam obtained from the experiment.

*3.3.2 Initial simulation of free-free beam:*

In machine learning, the larger the training set the more effective a neural network is trained. Nevertheless, collecting data from experiments requires a significant amount of cost and time. A useful and common alternative solution is to use a baseline FE model of the tested beam. The role of the FE model is to generate a training dataset based on many damage scenarios.

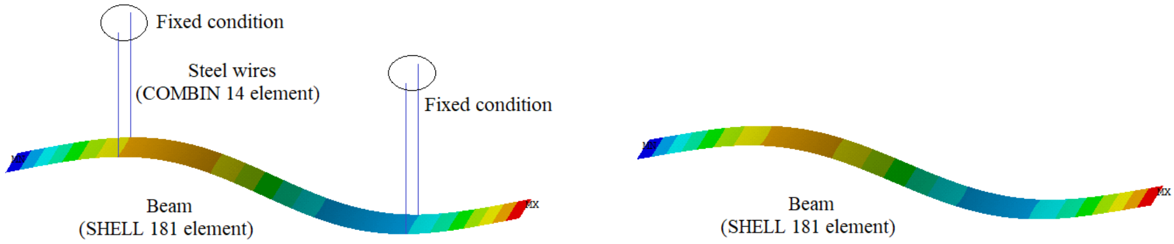
In a comparative study [39], three types of beam element, shell element, and solid element were used to investigate the convergence associated with variations of mesh sizes. The study indicates that the 2D model (using SHELL181 elements) showed better performance due to its simplicity, accuracy, and fast computation compared with other models. Therefore, the shell element is used in this study to build an initial finite element model (FEM) of the tested beam. The assumed initial values of material properties are Young's modulus $E$=1.98×10$^{11}$ N/mm$^2$, weight density $\gamma$ = 7850 kg/m$^3$, and Poisson's ratio $\nu$=0.3. SHELL181 element in ANSYS is used to model the beam with 16 elements. This element has four nodes each of which consists of six degrees of freedom (DOFs), i.e. three translations and three rotations about each axis (see Fig. 22b). Fig. 22a shows the constructed FE model of the beam in ANSYS.




a. FE model of the tested beam b. SHELL181 element

Fig. 22 Using shell 181 elements in simulation.

Two FE models were built for a preliminary comparison in order to choose a suitable simulation for model updating (see Fig. 23). The former used flexible springs, COMBIN14 element with a spring constant[2], $k_{vertical} = 10^5$ N/m, to simulate the steel wires. The latter was a free-free boundary condition model in ANSYS. The free-free boundary condition model had better agreement than the flexible spring model. The error between calculated and measured frequencies was smaller than that of the flexible-spring model (or cable-supported at two points 4 and 12). Besides, the higher MAC values in Equation (32) implied that the mode shapes of the free-free boundary condition model fitted the experimental ones. Therefore, the boundary conditions were considered free-free. The sensor weight was simulated by MASS21 element. Fifteen values of 0.0058kg, were placed at the corresponding sensor location along the tested beam. Block Lanczos method was employed for modal analysis [40].




a. Flexible-spring model b. Free-free boundary condition model

Fig. 23 Two comparative models for the experimental beam.

Values of natural frequencies as well as deviations in frequencies of the intact beam between simulation and measurement are shown in Table 7. The differences in the frequency values of the first five modes are under 0.45%. It means that the simulated data from the FE model matches the measured data very well.

In addition, for a more comprehensive evaluation, the similarity between two sets of mode shapes should be checked. A good correlation of the pairs of mode shapes guarantees the agreement between measurement and simulation. A common index, namely modal assurance criterion (MAC), is suggested for correlation checks. The computation of MAC values follows:

$$MAC_j = \frac{\left|\sum_{i=1}^{nn}(\phi_{i,j}^{measured})^T \times (\phi_{i,j}^{calculated})\right|^2}{\left(\sum_{i=1}^{nn}(\phi_{i,j}^{measured})^T \times (\phi_{i,j}^{measured})\right) \times \left(\sum_{i=1}^{nn}(\phi_{i,j}^{calculated})^T \times (\phi_{i,j}^{calculated})\right)} \tag{32}$$

where $i$=1 to $nn$ with $nn$ representing the number of degree of freedoms or measured points, $^T$ means transposition, $j$ implies the considered mode, $\phi_{i,j}^{measured}$ *and* $\phi_{i,j}^{calculated}$ represent respectively the simulated and experimental displacement mode shapes. From equation (32), MAC value equal to 1 indicates that the calculated mode shapes completely agree with the measured ones. In contrast, the unfitting between two mode shapes results in a value close to 0.

Diagonal MAC values of Table 7 show a superior agreement between calculated and measured displacement mode shapes because all values are over 0.9982. In other words, the vibration behavior of the initial FE model is close to the real one, although there are certain errors in natural frequency values. These discrepancies can be caused by the manufacturing of the beam, the mounting bases that are introduced in next section.

[2] The value of spring constant, $k_{vertical}$, was optimized based on the agreement with frequencies and mode shapes obtained from the measurement via an optimization process.

Table 7 Differences in frequency values and diagonal MAC values

| Mode | Measured frequency (Hz) | Calculated frequency (Hz) | Error(*) (%) | MAC Measurement - Calculation |
|---|---|---|---|---|
| 1 | 50.83 | 51.06 | 0.45 | 0.9995 |
| 2 | 140.4 | 140.76 | 0.26 | 0.9987 |
| 3 | 274.74 | 275.96 | 0.44 | 0.9991 |
| 4 | 456.94 | 456.18 | -0.17 | 0.9986 |
| 5 | 678.9 | 681.38 | 0.36 | 0.9982 |

Note: Error(*) = $(f_{calculated} - f_{measured}) \times 100 / f_{measured}$

*3.3.3 Model updating*

As mentioned in subsection 3.3.2, the FE model is used to generate data for training process. The accuracy of predicted outputs using experimental data depends significantly on this model. Although the initial FE modal results show a reasonable match to measured values, the errors need to be further reduced. To that end, updating of the initial FE model is applied to meet a good agreement between the calculated and measured modal parameters. FE model updating is a procedure of ensuring that the analysis results of the updated FE model are better reflections of the measured data compared to the initial model. In other words, this is a part of the verification and validation of the numerical model by varying some uncertain parameters that can affect the outputs of the FE model. The choice of the right updating parameters is crucial in this kind of problems. In general, uncertainties in material properties or boundary conditions can cause differences in natural frequencies. Regarding the boundary conditions, MAC values close to 1 indicate that the use of free-free boundary conditions in simulation is completely suitable. Therefore, the uncertainties in material properties are further considered in the process of model updating. These uncertainties are the value of Young's modulus and weight density of the steel beam. Moreover, to consider the imperfection due to the manufacture such as irregular cross-section, the mounting bases, beam segments were assigned different weights. The considered beam comprises 16 segments, therefore, there are sixteen parameters of weight density. Details of the updating parameters and ranges of changes are shown in Table 8.

Table 8 Updating parameters of the tested beam

| No | Updating parameters | Initial value | Lower bound | Upper bound |
|---|---|---|---|---|
| 1 | Young's Modulus, $E$ (N/mm$^2$) | $2.0\times10^{11}$ | $1.9\times10^{11}$ | $2.1\times10^{11}$ |
| 2 | Weight density, $\gamma_i$, $i$ = 1 to 16 (kg/m$^3$) | 7850 | 7750 | 8050 |

Another issue in model updating is the choice of objective (fitness) function. In this case, the differences in frequency values and displacement mode shapes are considered as the objective function of the marine predator algorithm (MPA). In other words, the objective of this step is to look for the minimum errors in frequencies and mode shapes between simulation and measurement. The optimization process using MPA is sketched in Fig. 24. From the flowchart, the process starts with random initial values of updating parameters. Then, these values are varied until the termination criteria are met. In this study, a maximum given iteration is the stopping criterion of the optimization process. In each iteration, the value of the objective function is recalculated as:

$$Objective\ function = \sum_{i=1}^{nm}\left(1-\frac{f_i^{calculated}\times f_i^{measured}}{\left(f_i^{measured}\right)^2}\right)^2+\sum_{i=1}^{nm}\left(1-MAC_i\right)^2 \tag{33}$$

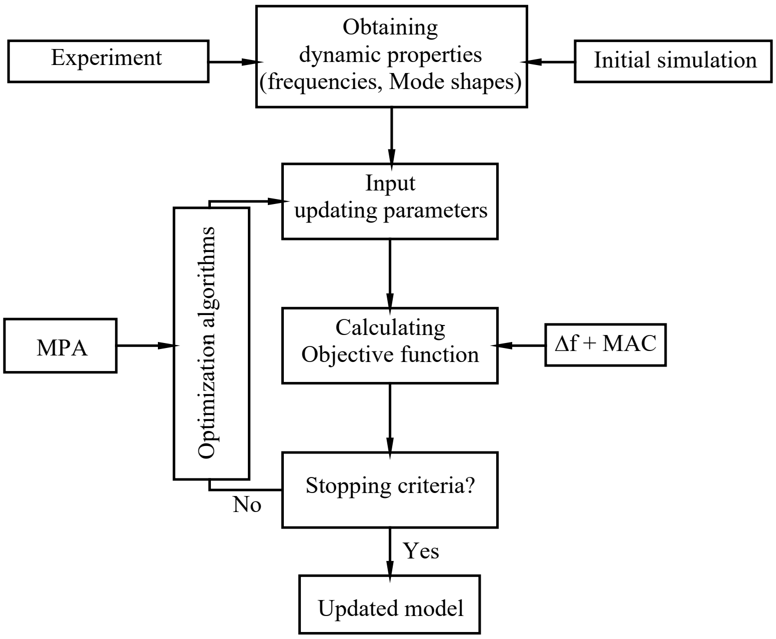



Fig. 24 Flowchart of the process of model updating using MPA.

There are some assumptions for MPA, namely distribution index for a stable process of Lévy, α=1.5, an index for FADs' effect or effect of Eddy formation on escapable ability from stagnancy in local optima, FADs=0.2, and a constant number for updating prey's position in the three main phases, P=0.5. In order to ensure the accuracy of optimized values, the population size of MPA is 50 and the maximum iteration equals 70. Fig. 25 shows the convergence curve of the best value of the objective function found in each iteration.

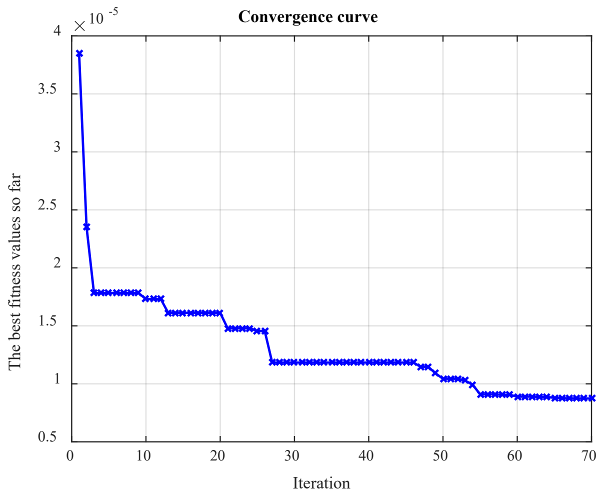



Fig. 25 Convergence curve of the best fitness function of marine predator algorithm

By inputting the new values of parameters in Table 10 to the initial model, Table 9 shows the modal properties of the updated FE model. It can be seen that all discrepancies in frequencies are significantly improved. The maximum error in frequency at the 4th mode is 0.13% while the error in the others is lower than 0.06%. The similarity between the two sets of mode shapes also performs a slight increase compared with the initial model. Therefore, the updated model can serve as a baseline model in the further step, damage identification.

Table 9 Comparison of frequency values before and after updating the FE model

| Mode | Initial model (Hz) | $Error_{ini}^{(*)}$ (%) | Updated model (Hz) | $Error_{up}^{(*)}$ (%) | $MAC_{updated}$ | Measurement (Hz) |
|---|---|---|---|---|---|---|
| 1 | 51.06 | 0.45 | 50.84 | 0.02 | 0.9995 | 50.83 |
| 2 | 140.76 | 0.26 | 140.38 | -0.01 | 0.9989 | 140.4 |
| 3 | 275.96 | 0.44 | 274.83 | 0.03 | 0.9991 | 274.74 |
| 4 | 456.18 | -0.17 | 456.35 | -0.13 | 0.9987 | 456.94 |
| 5 | 681.38 | 0.36 | 679.32 | 0.06 | 0.9984 | 678.9 |

Note: $Error^{(*)} = (f_{calculated} - f_{measured}) \times 100 / f_{measured}$

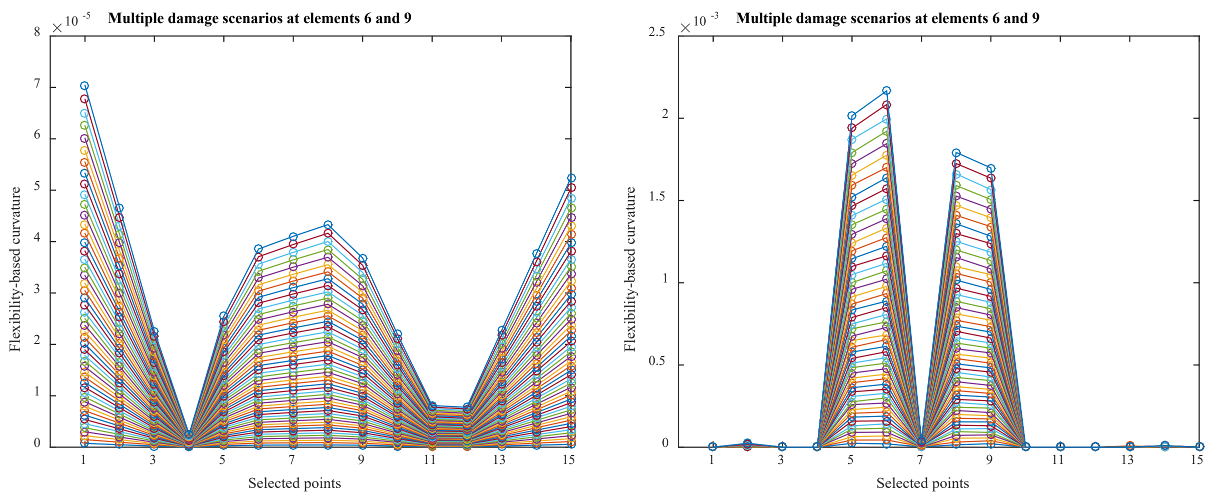
Table 10 Values of 17 uncertain parameters in initial FE model after updating

| $E$ | $\gamma_1$ | $\gamma_2$ | $\gamma_3$ | $\gamma_4$ | $\gamma_5$ | $\gamma_6$ | $\gamma_7$ | $\gamma_8$ |
|---|---|---|---|---|---|---|---|---|
| | 7830 | 8050 | 7750 | 7751 | 8035 | 7995 | 7750 | 7933 |
| $1.986\times10^{11}$ | $\gamma_9$ | $\gamma_{10}$ | $\gamma_{11}$ | $\gamma_{12}$ | $\gamma_{13}$ | $\gamma_{14}$ | $\gamma_{15}$ | $\gamma_{16}$ |
| | 8023 | 7970 | 7777 | 8050 | 7834 | 8050 | 7873 | 7758 |

*3.3.4 Application of MPAFNN in damage localization and quantification*

a. *Training procedure*

The main advantage of using modal flexibility is to be able to directly localize damages based on the change of flexibility matrix between the healthy and damaged state of the structure [30]. The flexibility-based curvature *MFC* is calculated, using the obtained experimental modal properties of the intact and damaged beam, from Equation (5). In the free-free condition structure, for a clearer view of damage position, a modification is proposed: $MFC = |\min(MFC, 0)|$ as depicted in Fig. 26. The figure indicates the location of the defect at nodes 6 and 9 with different damage extent. The maximum noise of 2% was added in the modal characteristics in both intact and damaged states to identify the *MFC* index.




a. Using flexibility changes b. Using modified flexibility-based curvature

Fig. 26 Damage localization using flexibility changes and the modified *MFC* index.

The modified flexibility-based curvature index can successfully indicate damage location in the free-free boundary condition beam. For an investigation purpose and reduction of computational time, eight elements within two nodes 4 and 12 (at two hanging positions of the beam) were chosen to generate damage scenarios. The stiffness variation, in this case, is from 1% to 50% with an interval of 1%. Changes in modal properties were used to compute the modified *MFC* index which is considered inputs of the FNN. The outputs of the FNN were label from 5 to 12 with their corresponding failure level. Therefore, to evaluate the effectiveness of the proposed method, there is a total of $N_{data} = 50 \times C_8^2 = 50 \times 28 = 1400$ samples. An FNN with the architecture 15-45-3 is trained with 80% of 1400 = 1120 samples and 858 predictors. The inputs of the FNN are the modified *MFC* indices while the outputs are two labels showing damaged element and a value indicating damage level. The same parameters of MPA are used as for this example, namely FADs=0.2, $P$=0.5, $\alpha$=1.5. A population of 500 is chosen. A maximum 20,000 iterations are set in settings to solve the damage detection problem.

*b. Testing the dataset*

The regression ability of MPAFNN was verified by using one-fifth of 1400 = 280 samples. Other four pairs of elements 6&7, 7&9, 7&11, 8&11, with various distributions in damage severities 35.5%, 40.3%, 32.5%, and 22.6%, were used to visually demonstrate the effectiveness of the proposed approach. The different noise levels created disturbances in the training data.

*c. Results of damage identification*

After 20,000 iterations, the obtained convergence curve of the FNN using MPA is plotted in Fig. 27. As its mission, the third phase continues to refine the search results. Regression plots based on the predictions and targets are performed in Fig. 28. They all perform superior results of fitting line, almost 45-angle line, as well as regression values, over 0.997. Therefore, it confirms the accuracy and reliability of MPAFNN in predicting values using its "learning ability".

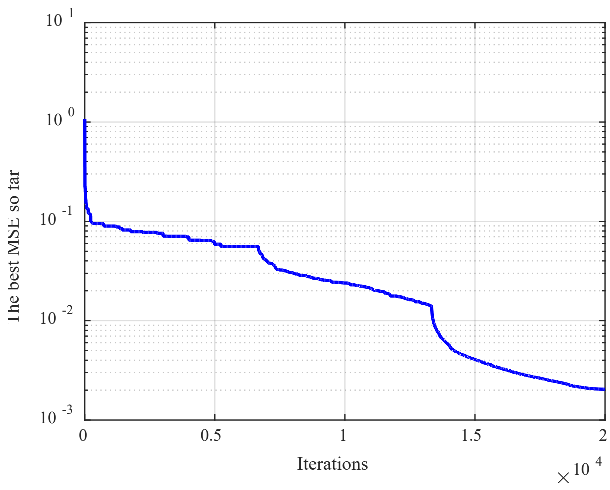
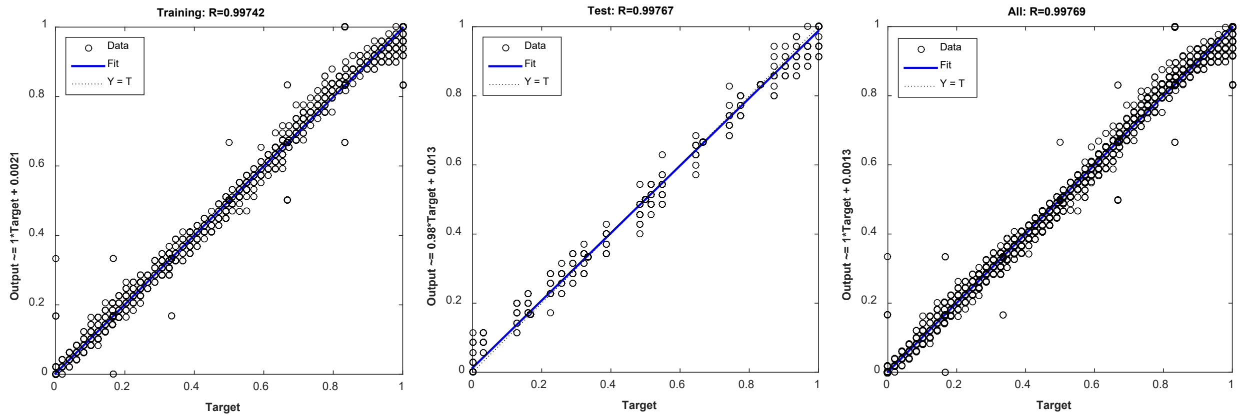



Fig. 27 Convergence curves for training process based on the best MSE of the objective function.

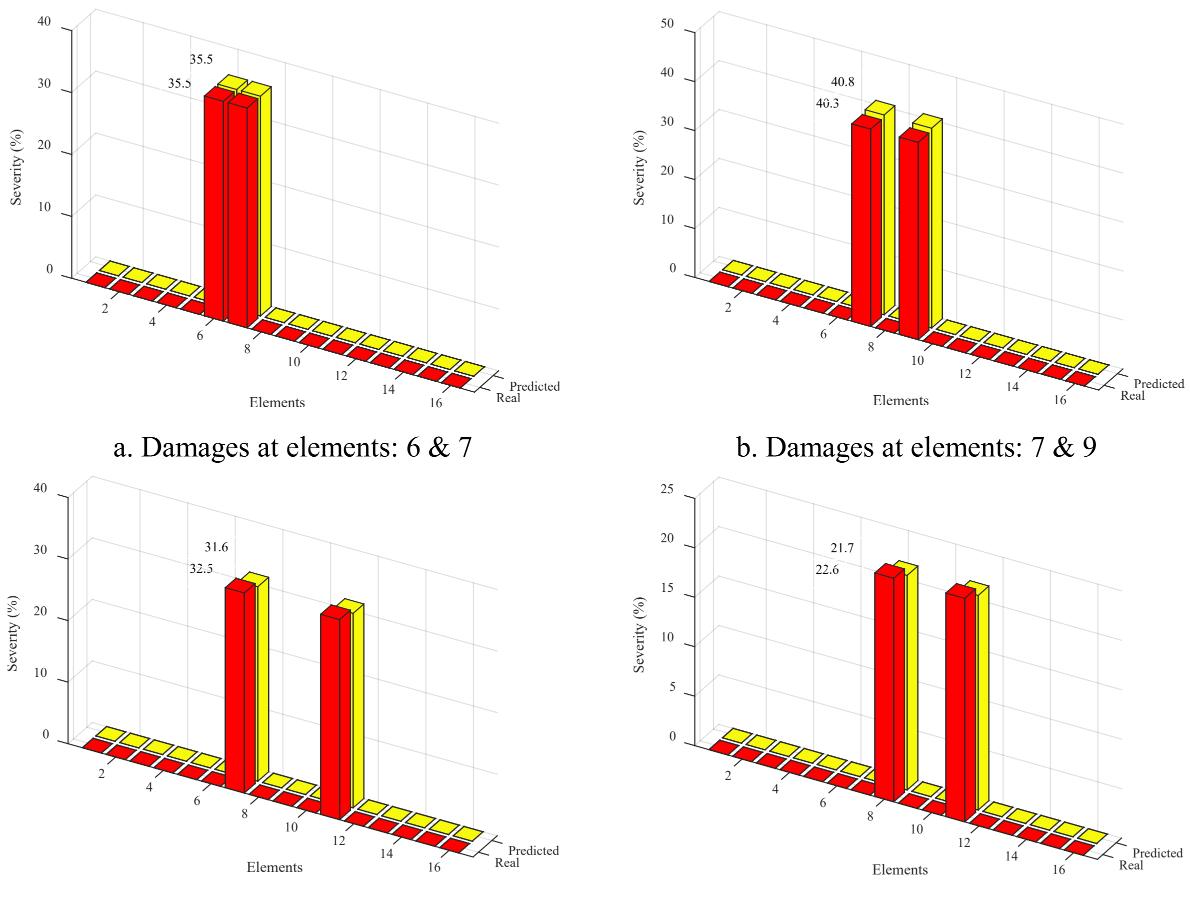
Based on the obtained weights and biases from the training process, the four damage cases were identified as in Fig. 29. The proposed method successfully determine all the damage location as well as corresponding extent even the training data was contaminated by different noise levels. However, it can be seen that the noise affected the accuracy of the severity prediction of MPAFNN. Excepting the first scenario, the others showed deviation between the real and predicted extent from 0.5% to 0.9%. The discrepancies are higher compared with the two previous cases without noise.




a. Training regression b. Test regression c. Overall regression

Fig. 28 The regression graphs using MPAFNN for the free-free beam

It has been noted that the results of damage identification have significantly improved using MPAFNN compared with using *MFC* indices alone confirming the fact that these indices are only useful in damage localization, and not for determining the extent of failure.




a. Damages at elements: 6 & 7 b. Damages at elements: 7 & 9

c. Damages at elements: 7 & 11 d. Damages at elements: 8 & 11

Fig. 29 Identification of multiple damages

## IV. Conclusions

A stochastic approach based on an optimization strategy is used to improve the learning ability of feedforward neural networks. The core of this approach is the nature-based optimization algorithm MPA. This algorithm is based on the foraging strategy, efforts in looking for the abundant environment and the marine memory of predator to find the optimal solution. The proposed combination of MPA and FNN, namely MPAFNN, has achieved some remarkable results in the prediction of both damage location and severity based on the training dataset:

- MPAFNN has superior performance regarding stability, reliability, precision and accuracy compared with GSAFNN, PSOFNN, PSOGSAFNN, GWOFNN. Results of average MSE, standard deviation MSE, and the best of MSE obtained from solving four classification benchmarks confirm the outstanding characteristics of MPAFNN.
- MPA is successful in solving the model updating problems based on measured data. The discrepancies between the natural frequencies of the first five modes obtained from the FE model and experiment are reduced significantly from 0.43% to 0.13% at the mode $4^{th}$ while the others are less than 0.06%.
- The main objective of MPAFNN, in this study, is to train a neural network to perform structural damage detection using measured experimental data. The obtained results demonstrate that MPAFNN can be used successfully for damage detection of various structures, i.e. simply supported beam, continuous beam, and free-free condition beam. MPAFNN successfully performs its prediction capability by mean of the regression plots. The good agreement between predicted and expected values from the training dataset, test dataset, and all dataset. Damage localization cases of single and multiple damage scenarios have been thoroughly handled by MPAFNN using either with or without noise in simulation data. The failure extent of the defective elements has been successfully updated over a series of damage cases. The maximum real error of all test cases in the numerical study is 0.9%.

Successful application of the proposed MPAFNN algorithm combining with *MFC* indices opens opportunities for using other modal damage indices for failure identification. We have shown that these indices can be further used for failure quantification.

## Acknowledgements

The authors acknowledge the financial support of VLIR-UOS TEAM Project, VN2018TEA479A103, 'Damage assessment tools for Structural Health Monitoring of Vietnamese infrastructures', funded by the Flemish Government.

vliruos SHARING MINDS, CHANGING LIVES With the support of THE BELGIAN DEVELOPMENT COOPERATION .be

This research is funded by University of Transport and Communications (UTC) under grant number T2021-PHII-006.

## Contributors

Long Viet Ho: Methodology, wrote the original draft of the manuscript, software, experiment; Duong Nguyen Huong: helped to organize the manuscript, experiment; Mohsen Mousavi: revised and organized the manuscript. Prof. Amir H. Gandomi: optimization algorithm; Prof. Thanh Bui-Tien, Prof. Magd Abdel Wahab, Prof. Guido De Roeck: supervision, and validation.

**Conflict of interest:** The authors declare no conflict of interest.

## References

[1] D. Maity and R. R. Tripathy, "Damage assessment of structures from changes in natural frequencies using genetic algorithm," *Structural Engineering and Mechanics*, vol. 19, no. 1, pp. 21–42, Jan. 2005. doi: 10.12989/sem.2005.19.1.021.

[2] H. Y. Guo and Z. L. Li, "A two-stage method to identify structural damage sites and extents by using evidence theory and micro-search genetic algorithm," *Mechanical Systems and Signal Processing*, vol. 23, no. 3, pp. 769–782, Apr. 2009, doi: 10.1016/j.ymssp.2008.07.008.

[3] J. D. Villalba and J. E. Laier, "Localising and quantifying damage by means of a multi-chromosome genetic algorithm," *Advances in Engineering Software*, vol. 50, pp. 150–157, Aug. 2012, doi: 10.1016/j.advengsoft.2012.02.002.

[4] M. I. Friswell, J. E. T. Penny, and S. D. Garvey, "A combined genetic and eigensensitivity algorithm for the location of damage in structures," *Computers & Structures*, vol. 69, no. 5, pp. 547–556, Dec. 1998, doi: 10.1016/S0045-7949(98)00125-4

[5] M. Nobahari and S. M. Seyedpoor, "Structural damage detection using an efficient correlation-based index and a modified genetic algorithm," *Mathematical and Computer Modelling*, vol. 53, no. 9–10, pp. 1798–1809, May 2011, doi: 10.1016/j.mcm.2010.12.058.

[6] T. N. Hoa and B. T. Thanh, "Damage Detection In A Steel Beam Structure Using Particle Swarm Optimization And Experimentally Measured Results," *Science Journal of Transportation*. **09**: p3-9.

[7] S. S. Nanthakumar, T. Lahmer, X. Zhuang, G. Zi, and T. Rabczuk, "Detection of material interfaces using a regularized level set method in piezoelectric structures," *Inverse Problems in Science and Engineering*, vol. 24, no. 1, pp. 153–176, Jan. 2016, doi: 10.1080/17415977.2015.1017485.

[8] O. Salawu and C. Williams, "Bridge Assessment Using Forced-Vibration Testing," *Structural Engineering*, vol. 121, no. 2, pp. 161–173, Feb. 1995, doi: 10.1061/(ASCE)0733-9445(1995)121:2(161).

[9] J.-M. Ndambi, J. Vantomme, and K. Harri, "Damage assessment in reinforced concrete beams using eigenfrequencies and mode shape derivatives," *Engineering Structures*, vol. 24, no. 4, pp. 501–515, Apr. 2002, doi: 10.1016/S0141-0296(01)00117-1.

[10] A. Khatir, M. Tehami, S. Khatir, and M. Abdel Wahab, "Republished Paper. Multiple damage detection and localization in beam-like and complex structures using co-ordinate modal assurance criterion combined with firefly and genetic algorithms," *J VIBROENG*, vol. 20, no. 1, pp. 832–842, Feb. 2018, doi: 10.21595/jve.2016.19719.

[11] S. H. Petro, S.-E. Chen, H. V. S. GangaRao, and S. Venkatappa, "Damage Detection Using Vibration Measurements," *Proceedings of SPIE - The International Society for Optical Engineering*, p. p.113-119, Jan. 1997.

[12] A. Alvandi and C. Cremona, "Assessment of vibration-based damage identification techniques," *Journal of Sound and Vibration*, vol. 292, no. 1–2, pp. 179–202, Apr. 2006, doi: 10.1016/j.jsv.2005.07.036.

[13] A. Entezami and H. Shariatmadar, "Damage detection in structural systems by improved sensitivity of modal strain energy and Tikhonov regularization method," *Int. J. Dynam. Control*, vol. 2, no. 4, pp. 509–520, Dec. 2014, doi: 10.1007/s40435-014-0071-z.

[14] M. Montazer and S. M. Seyedpoor, “A New Flexibility Based Damage Index for Damage Detection of Truss Structures,” *Shock and Vibration*, vol. 2014, pp. 1–12, 2014, doi: 10.1155/2014/460692.

[15] S. A. Bagherahmadi and S. M. Seyedpoor, “Structural damage detection using a damage probability index based on frequency response function and strain energy concept,” *Structural Engineering and Mechanics*, vol. 67, no. 4, pp. 327–336, Aug. 2018, doi: 10.12989/SEM.2018.67.4.327.

[16] A. Behtani *et al.*, “The Sensitivity of Modal Strain Energy for Damage Localization in Composite Stratified Beam Structures,” in *Proceedings of the 13th International Conference on Damage Assessment of Structures*, M. A. Wahab, Ed. Singapore: Springer Singapore, 2020, pp. 863–874.

[17] M. M. Abdel Wahab and G. De Roeck, “Damage Detection In Bridges Using Modal Curvatures: Application To A Real Damage Scenario,” *Journal of Sound and Vibration*, vol. 226, no. 2, pp. 217–235, Sep. 1999, doi: 10.1006/jsvi.1999.2295.

[18] R. Salgado, P. Cruz, L. Ramos, and P. Lourenço, “Comparison between damage detection methods applied to beam structures,” in *Bridge Maintenance, Safety, Management, Life-Cycle Performance and Cost*, Porto, Portugal, Jul. 2006, pp. 241–242, doi: 10.1201/b18175-86.

[19] P. J. S. Cruz and R. Salgado, “Performance of Vibration-Based Damage Detection Methods in Bridges,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, no. 1, pp. 62–79, Jan. 2009, doi: 10.1111/j.1467-8667.2008.00546.x.

[20] Wei Fan and Pizhong Qiao, “Vibration-based Damage Identification Methods: A Review and Comparative Study,” *Structural Health Monitoring*, vol. 10, no. 1, pp. 83–111, Jan. 2011, doi: 10.1177/1475921710365419.

[21] I. Talebinejad, C. Fischer, and F. Ansari, “Numerical Evaluation of Vibration-Based Methods for Damage Assessment of Cable-Stayed Bridges: Numerical evaluation of vibration-based methods,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 3, pp. 239–251, Apr. 2011, doi: 10.1111/j.1467-8667.2010.00684.x.

[22] V. B. Dawari and G. R. Vesmawala, “Modal Curvature and Modal Flexibility Methods for Honeycomb Damage Identification in Reinforced Concrete Beams,” *Procedia Engineering*, vol. 51, pp. 119–124, 2013, doi: 10.1016/j.proeng.2013.01.018.

[23] Wickramasinghe, Wasanthi Ramyalatha, Thambiratnam, David, & Chan, Tommy (2015) Use of modal flexibility method to detect damage in suspended cables and the effects of cable parameters. *Electronic Journal of Structural Engineering*, 14(1), pp. 133-144.

[24] C. Anitescu, E. Atroshchenko, N. Alajlan, and T. Rabczuk, “Artificial Neural Network Methods for the Solution of Second Order Boundary Value Problems,” *Computers, Materials & Continua*, vol. 59, no. 1, pp. 345–359, 2019, doi: 10.32604/cmc.2019.06641.

[25] E. Samaniego et al., “An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications,” *Computer Methods in Applied Mechanics and Engineering*, vol. 362, p. 112790, Apr. 2020, doi: 10.1016/j.cma.2019.112790.

[26] T. N. Hoa, S. Khatir, G. De Roeck, N. N. Long, B. T. Thanh, and M. A. Wahab, “An efficient approach for model updating of a large-scale cable-stayed bridge using ambient vibration measurements combined with a hybrid metaheuristic search algorithm,” *Smart Structures and Systems*, vol. 25, no. 4, pp. 487–499, Apr. 2020, doi: 10.12989/SSS.2020.25.4.487.

[27] H. Tran-Ngoc *et al.*, “An efficient approach to model updating for a multispan railway bridge using orthogonal diagonalization combined with improved particle swarm optimization,” *Journal of Sound and Vibration*, vol. 476, p. 115315, Jun. 2020, doi: 10.1016/j.jsv.2020.115315.

[28] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, “Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems,” *Computers & Structures*, vol. 110–111, pp. 151–166, Nov. 2012, doi: 10.1016/j.compstruc.2012.07.010.

[29] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine Predators Algorithm: A nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, p. 113377, Aug. 2020, doi: 10.1016/j.eswa.2020.113377.

[30] A.K. Pandey, M. Biswas, "Damage Detection in Structures Using Changes in Flexibility," *Journal of Sound and Vibration*, Volume 169, Issue 1, 1994, Pages 3-17, https://doi.org/10.1006/jsvi.1994.1002.

[31] F. N. Catbas, M. Gul and J. L. Burkett "Damage assessment using flexibility and flexibility-based curvature for structural health monitoring", Smart Material and Structures, 17 (2008) 015024.

[32] S. Mirjalili, S. Z. Mohd Hashim, and H. Moradian Sardroudi, “Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm,” *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125–11137, Jul. 2012, doi: 10.1016/j.amc.2012.04.069.

[33] L. V. Ho, D. H. Nguyen, G. D. Roeck, T. Bui-Tien, and M. A. Wahab, “Damage detection in steel plates using feed-forward neural network coupled with hybrid particle swarm optimization and gravitational search algorithm,” *Journal of Zhejiang University-Science A (Applied Physics & Engineering)*, p. 14, 2020, doi: 10.1631/jzus.A2000316

[34] S. Mirjalili, “How effective is the Grey Wolf optimizer in training multi-layer perceptrons,” *Appl Intell*, vol. 43, no. 1, pp. 150–161, Jul. 2015, doi: 10.1007/s10489-014-0645-7.

[35] B. Peeters and G. De Roeck, “Stochastic System Identification for Operational Modal Analysis: A Review,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 4, pp. 659–667, Dec. 2001, doi: 10.1115/1.1410370.

[36] MACEC 3.2: A Matlab toolbox for experimental and operational modal analysis.

[37] B. Peeters and G. De Roeck, “Reference-based stochastic subspace identification for output-only modal analysis,” *Mechanical Systems and Signal Processing*, vol. 13, no. 6, pp. 855–878, Nov. 1999, doi: 10.1006/mssp.1999.1249.

[38] B. Peeters, “System Identification and Damage Detection in Civil Engineering”, Ph.D. Dissertation; Katholieke Universiteit Leuven, Belgium” p. 256, 2000.

[39] D. H. Nguyen, L. V. Ho, T. Bui-Tien, G. De Roeck, and M. A. Wahab, “Damage Evaluation of Free-Free Beam Based on Vibration Testing,” *Applied Mechanics*, vol. 1, no. 2, pp. 142–152, May 2020, doi: 10.3390/applmech1020010.

[40] ANSYS, Inc. Southpointe, 275 Technology Drive, Canonsburg, PA 15317, Release 17.2.