



# Deep Neural Network and YUKI Algorithm for Inner Damage Characterization Based on Elastic Boundary Displacement

Nasreddine Amoura<sup>1</sup>, Brahim Benaissa<sup>2</sup>(✉), Musaddiq Al Ali<sup>3</sup>, and Samir Khahir<sup>4</sup>

<sup>1</sup> LMP2M Laboratory, Yahia Fares University of Medea, New Urban Pole, 26000 Médéa, Algeria

amoura.nasreddine@univ-medea.dz

<sup>2</sup> Design Engineering Lab, Department of Mechanical Systems Engineering, Toyota Technological Institute, 2-12-1, HisakataNagoya, Aichi 468-8511, Tenpaku-Ku, Japan  
Benaissa@toyota-ti.ac.jp

<sup>3</sup> Department of Advanced Science and Technology, Toyota Technological Institute, 2-12-1, HisakataNagoya, Aichi 468-8511, Tenpaku-Ku, Japan

<sup>4</sup> Ho Chi Minh City Open University, Ho Chi Minh, Viet Nam

**Abstract.** The efficiency of deep neural networks has been proven in several research fields. In this study, we suggest using this method of inverse crack identification based on the structural response of boundary displacement. This structural response is particularly challenging for surrogate modelling due to the overall similarity in the effect of different cracks. From the inverse problem perspective, this corresponds to a problem of many local minima. To solve this problem we use the newly suggested search technique of the dynamic search space reduction by the YUKI algorithm, build to solve this type of problem. We compare the performance of the suggested approach of the RBF modelling technique in terms of direct problem prediction and inverse problem identification accuracy. Deep Neural Networks are found to have better performance in both problems, although the computation time is significantly higher than RBF.

**Keywords:** YUKI algorithm · Crack identification · Deep ANN · Inverse problem

## 1 Introduction

In the framework of inverse crack identification the goal is to predict the unknown crack, defined by its parameters  $P$  [1, 2], based only on accessible structural response measurements  $u(P_0)$ . The most commonly used are the vibrational response and the mechanical responses. However, they face the challenge of the ill-Posedness [3–8], as the measured structural response may be insensitive to different cracks, and very different values of  $P$  can correspond to close measurements. Additionally, from the optimization problem perspective, this is a non-differentiable problem.

The crack identity  $P$  is dictated by the considered inverse problem. Thus, the correct numerical simulation of the problem is very important. The second difficulty is related

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023

R. Capozucca et al. (Eds.): ICSCES 2022, LNCE 317, pp. 220–233, 2023.

[https://doi.org/10.1007/978-3-031-24041-6\\_18](https://doi.org/10.1007/978-3-031-24041-6_18)

to structural response characteristic in relation to variation to the crack parameters [9]. The existence of local minima is inevitable in these type of problems. Where local optimization algorithms cannot guarantee to attain the global minimum. Plus they require the problem to be differentiable. For these reasons, global optimization algorithms are needed [3, 10, 11].

An efficient algorithm is capable of finding the global minimum, regardless of the starting point in the search field. Global methods simulate a sample of parameters by exploiting only the information provided by the calculation of  $f(P)$ . To get the global minimum of a function using global methods, it would be necessary to go through all the space of the independent variables, which is generally impossible for classical optimization methods, given the size of the space and the number of variables.

In the inverse problem stage, the optimization algorithm is applied. Its goal is to find the closest response to one consequence of the crack to be identified [6]. The structural response is measured at the reference specimen. The optimization algorithm will then search for the crack parameters that correspond to the measured response. By calculating the structural response  $u(P)$  corresponding to a possible crack  $P$ . The fitness function value is the error between this vector and the reference structural response  $u(P_0)$  caused by the real crack parameters. [12]. Figure 1 depicts the inverse crack identification approach. Where the optimal crack identity is the one that provides a fitness function value less or equal to a limit. This limit varies from one case to another and also on the nature of the crack parameters. It is decided after knowing the fitness function equivalent to the wanted precision. The stopping criterion is considered based on the number of iterations. The number of iterations depends on the optimization algorithm, so the optimal crack identity can be reached if the optimization algorithm is fast enough.

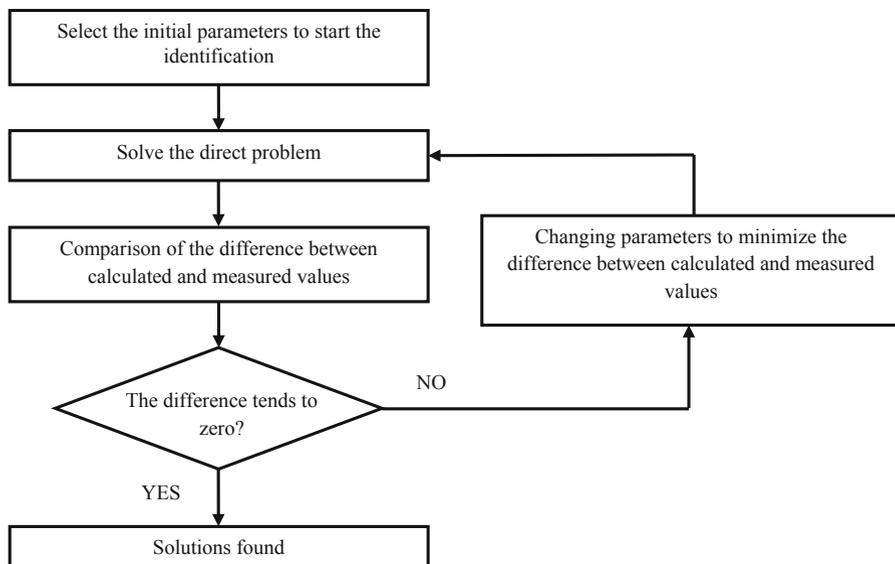


Fig. 1. Inverse crack identification approach

The structural response researchers have suggested several approaches based on ANN [13], Weinstein et al. suggested. One of the early deep learning works in this field of vibration analysis is made by Barai et al. [14] in damage identification Zang et al. [15] suggested an approach based on frequency response functions as input to ANN. Abdeljaber et al. [16] designed a real-time damage identification strategy based on one-dimensional convolutional neural networks. In the studies based on the mechanical response, Stavroulakiset al [17, 18] created a crack identification method based on backpropagation neural networks for structural response generation, using the boundary element method to create the training data. Khaleghi et al. [19] suggested an inner fracture characteristics approach based on artificial neural networks under noisy measurement.

## 2 Numerical Simulation

In elastostatics, the sum of all forces on the structure is equal to zero, and the displacements are not a function of time [20–22]. The equilibrium equations are stated as:

$$\sigma_{ij,j} + b_i = 0, \quad i, j = 1, 2, 3 \tag{1}$$

$\sigma_{ij}$  is the Cauchy stress tensor.  $b_i$  represents the body-force components. And the strains  $\varepsilon_{ij}$  are defined by:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}), \quad i, j = 1, 2, 3 \tag{2}$$

$u_{i,j}$  are the displacement component. Considering that the material is elastic, homogenous and isotropic, Eqs. (1) and (2) are related by Lamé’s equation:

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \delta_{ij}, \quad i, j, k = 1, 2, 3 \tag{3}$$

The Kronecker-delta function  $\delta_{ij}$  is written as follows:

$$\delta_{ij} = \{1 \quad i = j, \quad 0 \quad i \neq j\} \tag{4}$$

$$\lambda = \frac{2\nu\mu}{(1 - 2\nu)} \tag{5}$$

$$\mu = \frac{E}{2(1 + \nu)} \tag{6}$$

where  $\lambda$  Lamé constant, and  $\mu$  is the shear modulus. The strains have to satisfy the following compatibility equations:

$$\frac{\partial \varepsilon_{ij}}{\partial x_i \partial x_k} - \frac{\partial}{\partial x_i} \left( -\frac{\partial \varepsilon_{jk}}{\partial x_i} + \frac{\partial \varepsilon_{ik}}{\partial x_j} + \frac{\partial \varepsilon_{ij}}{\partial x_k} \right) = 0, \quad i, j, k = 1, 2, 3 \text{ and } i \neq j \neq k \tag{7}$$

The problem is defined as follows. Let  $\Omega$  denote an open set with boundary  $\Gamma$  subject to the boundary conditions:

$$u = \underline{u}_i \text{ on } \Gamma_u, \tag{8}$$

$$t = \underline{t}_i \text{ on } \Gamma_t \tag{9}$$

Where

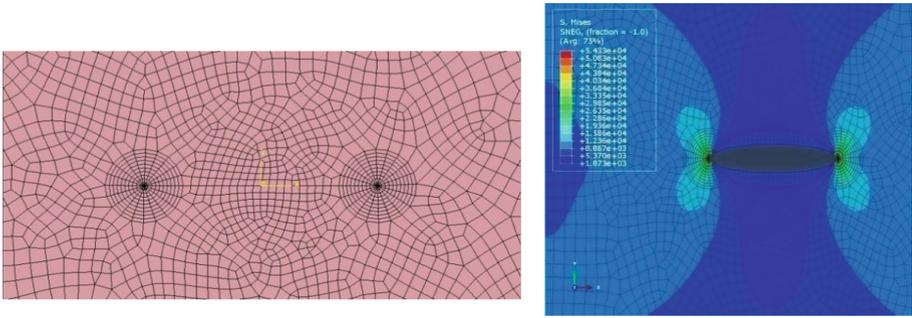
$$\Gamma = \Gamma_u \cup \Gamma_t, \Gamma_u \cap \Gamma_t = \emptyset \tag{10}$$

And

$$\underline{\Omega} = \Omega \cup \Gamma \tag{11}$$

The problem is: given  $u_i$ ,  $t_i$  and  $b_i$ ; find  $u_i(y)$ ,  $y \in \underline{\Omega}$ . Furthermore, the plane strain is assumed and therefore:

$$\varepsilon_{zz} = \varepsilon_{xz} = \varepsilon_{yz} = 0 \tag{12}$$



**Fig. 2.** Display of meshing around the crack tips and Von Mises constraint

The simulations in this paper are made using Abaqus. It is a powerful finite element simulation tool used in many fields of mechanical studies. It has special features dedicated to fracture mechanics, which makes this tool very advantageous, such that the crack segment model, the calculation of J integral, and different crack propagation laws.

In this section, we present a simulation of a cracked plate of a rectangular shape with 1 mm thickness, having dimensions of 30 mm height and 10 mm width, it is subjected to traction force from the upper and the lower sides. The Young modulus and Poisson coefficient of the material were respectively  $E = 210 \text{ GPa}$  and  $\nu = 0.3$ . The vertical sides have meshed with 80 quadrilaterals and the horizontal side with 30 elements. The meshing of the crack has been divided into two main areas; the area of the crack tip, covering both crack endings, and the area of the middle of the crack. The sweep function is used to mesh the crack tips, with radius of the circle is 0.5 mm. 20 elements along the contour of the circle, and 10 elements along the radius of the circle, resulting in a total of 200 elements at each crack tip as shown in Fig. 2. Along with the the Von Mises constrain.

The majority of crack identification studies are based on boundary data because boundaries are considered the only accessible part of the structure. In the case studied

in this paper, the horizontal boundaries (upper and lower boundaries) are subjected to boundary conditions, in real cases; it means interaction with other structures of solicitations, which makes them inaccessible for measurements, leaving the vertical boundaries to be the only source of information.

### 3 Radial Basis Functions Modelling

Radial Basis Functions (RBF) are very efficient for interpolation between existing data [23, 24]. To approximate a function  $f(x)$ , where  $x$  is an  $M$ -dimensional vector, RBF requires the information in a form of a set of  $N$  nodes  $x_i$ , for which the values of the function are known. For any new value of  $x$  the interpolation is performed involving just a few nearby nodes. The approach of RBF performs it one continuous function, defined over the whole domain. The approximation is written as the combination of  $g_i$  functions:

$$f(x) \approx \sum_{i=1}^N \alpha_i g_i(x) \quad (13)$$

This equation is defined once the basis functions  $g_i$  are selected and the coefficients  $\alpha_i$  are known. Where  $\alpha_i$  are the combination coefficients. For the radial basis functions, the Euclidian distance is considered in this paper.

$$g_i(x) = g_i(\|x - x_i\|), i = 1, 2, \dots, N \quad (14)$$

To determine the coefficients  $\alpha_i$ , the interpolation needs to be exact in all  $N$  nodes, therefore, the system of  $N$  equations defined by:

$$f(x_j) = y_j = \sum_{i=1}^N \alpha_i g_i(x_j), j = 1, \dots, N \quad (15)$$

where  $y_j$  are known values of the function in the nodes. Introducing the following matrix notation

$$G = \begin{bmatrix} g_1(x_1) & \dots & g_N(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_N) & \dots & g_N(x_N) \end{bmatrix}; \alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T; Y = [y_1, y_2, \dots, y_N]^T \quad (16)$$

The system can be written as:

$$\alpha \cdot G = Y \quad (17)$$

Matrix Eq. (15) is solved for the interpolation coefficients  $\alpha_i$ . The interpolation coefficients are computed once and for all, and involve the known values at the nodes. Once this is achieved, we can approximate the function at any given point [23]. Considering that the results of Eq. (13) are exact in the nodes, and gives interpolated results for any new value of  $x$ . The RBF represents one approximation valid for the whole domain in which the original data were situated.

The results is a model that is able of reproducing the original data field, plus interpolated values for any new set of parameters within the initial variables domain. The results of extrapolation outside this domain may be of poor accuracy [23]. We use the RBF model to retrieve the values of the unknown geometric crack parameters in this study. And employed to provide the structural response, representing the direct model.

### 4 Deep Artificial Neural Network

Deep ANN has been used widely in a wide range of engineering problems [25, 26]. Theoretically, it has the power to connect the problem’s input and target features in nonlinear and complex spaces, given enough neurons in the hidden layers. With the wide availability of ANN tools and open source codes, engineers only have to solve the issue of the number of layers, and the number of neurons in each layer [27].

The information is stored in the node’s weights. Thus for the network to approximate the system’s output, these weights are optimized in a process called training. Figure 3. Illustrates the Deep ANN network. Training of collected data is used to model the characteristics of the system. During the training, the process uses a smaller set of data called the validation dataset is used test the quality of the network predictions, and change the parameters to achieve high precision [28].

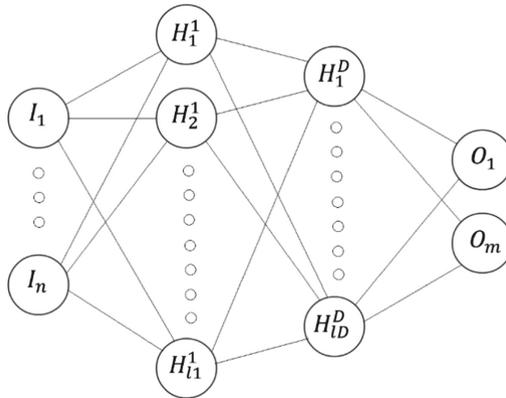


Fig. 3. Von Mises constraint ANN structure to determine damage level

The Deep ANN is characterized by multiple hidden layers, besides the input layer and the output layer. The number of hidden layers is denoted in the figure by  $D$ , the number of input and output neurons are denoted by  $n$  and  $m$  respectively, and the number of neurons in each hidden layer is denoted by  $l$ . Each neuron is connected to all the neurons in the next layer, and the strength of the connection is weighted though  $w_{ij}^k$ . Each neuron is assigned a bias value  $\beta$ . In the training phases, the optimization algorithm will find the optimal weights and biases combinations that correspond to the lowest prediction error, according to the testing data set.

### 5 YUKI Algorithm

YUKI algorithm is a newly suggested method [20], with an innovative search space reduction technique. It uses simple steps to focus on interesting search areas, and dynamically changes the size and position of the search focus. The first idea of YUKI algorithm is to create a local search area smaller than the search domain, it is positioned around

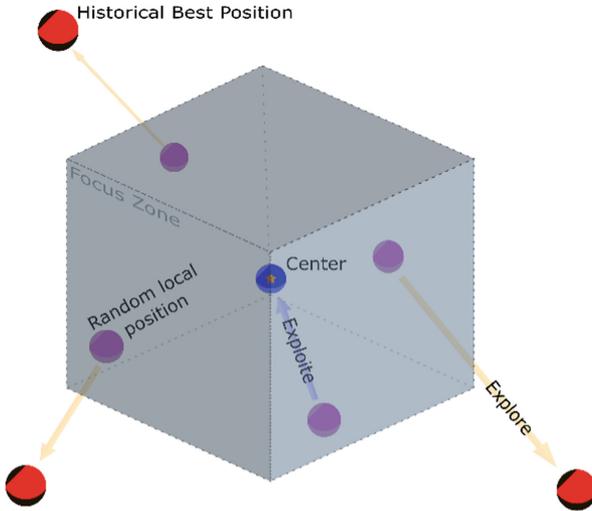
the absolute best solution at the current iteration  $X_{best}$ , and its size is calculated as the distance between this point and the  $X_{MeanBest}$ . Where MeanBest is the centre of the point cloud representing the historical best founts found by each solution.  $LT$  and  $LB$  are the top and bottom boundaries of the local search area respectively. They are calculated as follows:

$$D = |X_{best} - X_{MeanBest}| \tag{1}$$

$$LT = X_{best} + D \tag{2}$$

$$LB = X_{best} - D \tag{3}$$

The second idea of this algorithm is to dedicate two parts of the population to exploration and exploitation simultaneously, by splitting the population. One part is assigned to exploration outside the local search area, and the other is assigned to focus on searching inside the local search area. The size of the population dedicated to each part is determined by the EXP parameter, which should be between 0 and 1. 0.6 for example, means 60% of the population will be dedicated to exploring outside the focus area, and 40% will focus the search inside the local search area.



**Fig. 4.** Illustration of the focus zone exploration strategy

This is done by first, generating a random distribution of points inside the local search area, then assigning the dedicated search populations selected for exploration outside the local search area. Exploration spread is directed to look toward the historical best points as follows:

$$E = X_{loc} - X_{best} \tag{5}$$

where  $X_{loc}$  denotes a position of the selected point to be directed toward exploration, and  $X_{best}$  is the historical best point found by this individual particle. The next position will then be calculated as follows:

$$X_{new} = X_{loc} + E \tag{6}$$

For the solutions dedicated to the exploitation inside the local search zone, we first calculate the distance from the centre Eq. (7), then calculate the next position as in Eq. (8). Where  $rand$  here is the random value between 0 and 1. And for each solution. Figure 4 illustrates the YUKI algorithm ideas.

$$F = X_{loc} - X_{best} \tag{7}$$

$$X_{new} = X_{loc} - rand \times F \tag{8}$$

These simple steps lead to the complex behaviour of the local search area, as it can dynamically reduce its size if new solutions are found inside, or increase the size significantly if better solutions are found outside. The location of this focus area also changes automatically throughout the search. With the progress of iterations, the local search area tends to reduce in size leading to focusing on the optimal position at the end of the search.

### 6 ANN and RBF for Direct Problem Representation

The boundary displacement responses are collected from 67 simulations. The boundary conditions are constant but crack sizes and positions vary. The size of the crack varies between 0 and 5 mm, and its position is located between -10 and 10 on the Y-axis and -2 and 2 on the X-axis (Fig. 5). Each node in the vertical boundaries is considered a sensor point, thus the total number of 162 data points, 81 from the left side and 81 from the right vertical side boundary.

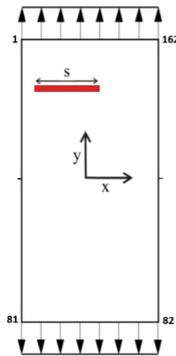


Fig. 5. Illustration of the simulated problem with crack parameters.

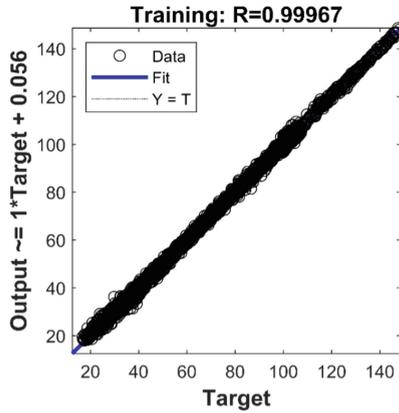


Fig. 6. Displays the training results for the deep ANN network.

These results are used to build surrogate models using RBF and ANN networks. Several ANN designs have been investigated. After statistical analysis, the backpropagation architecture with these layers (60:150:250:200:180) has been selected for further study. Figure 6 displays the training results for this deep ANN network, using conjugate gradient backpropagation with Fletcher-Reeves updates.

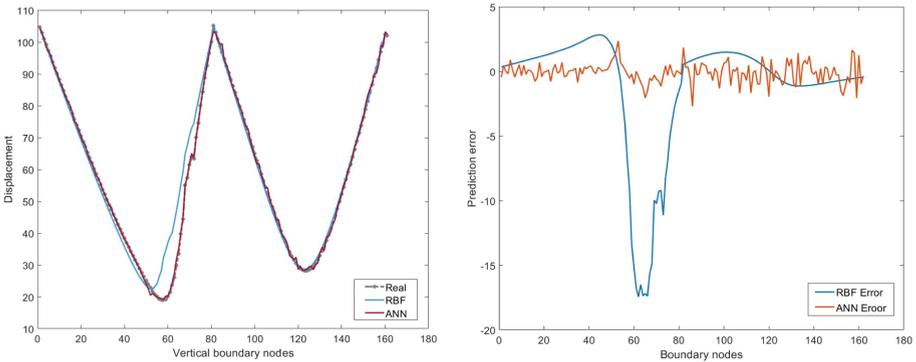


Fig. 7. Boundary displacement prediction error, using deep ANN and RBF networks.

Figure 7 Shows the boundary displacement prediction error using deep ANN and RBF. Where the considered example is a crack at the lower left side with center coordinates  $(-1.75, -9.5)$  and a size of 3.2 mm. The left side graph compares the predictions to of the real displacement field. And on the right side graph the prediction errors of deep ANN and RBF. These figures show an advantage for Deep ANN in terms of overall accuracy, it also indicates that although RBF has a smoother displacement field, more appropriate in this case than the results found by deep ANN, RBF faces a challenge in predicting the displacement at that boundary node in the proximity of the crack. These results are found in other examples and are indicative of the overall predictions.

## 7 Crack Identification Results

The crack identification is formulated as an optimization problem, as suggested in Fig. 1, the goal is to find the crack corresponding to the boundary displacement response most close to the measurement, the closeness is calculated as described in Eq. (9). We considered 4 segments of the displacement field, representing the four quarters of the vertical boundaries.

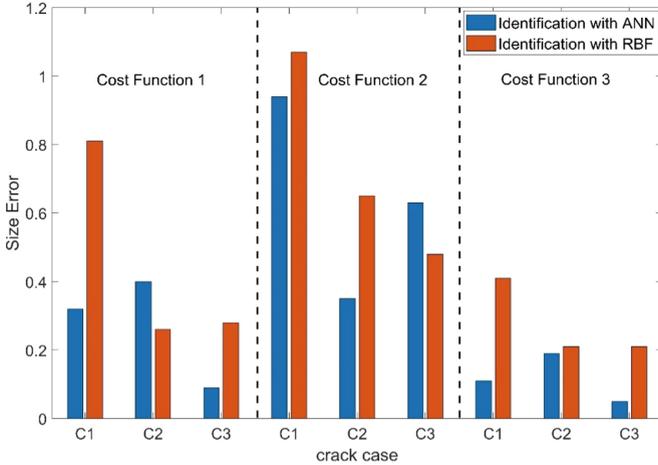


Fig. 8. Crack size prediction based on boundary displacement using Deep ANN and RBF.

Due to the unsymmetric nature of the problem, we assume that this approach help identify the region in which the crack exists more efficiently than if we consider the boundary displacement field as a whole. Three cost functions were considered, expressed in Eqs. (10), (11) and (12).

$$f^i(P) = \frac{\|u^i(p_0) - u^i(p)\|^2}{\|u^i(p_0)\|^2} \text{ with } i = 1, 2, 3, 4. \tag{9}$$

$$F_1(P_{opt}) = \min(f^1(P), f^2(P), f^3(P), f^4(P)) \tag{10}$$

$$F_2(P_{opt}) = \text{Mean}(f^1(P), f^2(P), f^3(P), f^4(P)) \tag{11}$$

$$F_2(P_{opt}) = F_1 + F_2 \tag{12}$$

The first function takes into consideration only the boundary quarter corresponding to the minimum fitness value. This is designed to isolate the noisy effect created by the rest of the data points. The second function is similar to the case where we consider the whole boundary displacements, it has the advantage of having better knowledge of the effect of each particular crack on the boundary displacement field.

The last function takes into consideration the advantages of both functions, and it is assumed in this case, that the fitness of wrong crack identities will be amplified due to having the error considered twice. Tables 1, 2 and 3 show the identification errors in terms of crack size and position, the error is calculated as the Euclidian distance between the real crack centre and the centre of the predicted crack. The first crack is C1 with centre coordinates (-1.7, -9.5) and a size of 3.2 mm, the C2 centre is at (1.6, 3.8) and a size of 3.1 mm. C3 is positioned at (-0.15, 0.15) and has a size of 4.6 mm.

**Table 1.** Identification results based on the first fitness function (Eq. 10).

Fitness Function $F_1$	ANN Error (mm)		RBF Error (mm)	
	Size	Position	Size	position
C1	0.32	0.877268	0.81	1.203703
C2	0.4	0.297321	0.26	0.766877
C3	0.09	0.632456	0.28	0.764003

**Table 2.** Identification results based on the second fitness function (Eq. 11).

Fitness Function $F_2$	ANN Error (mm)		RBF Error (mm)	
	Size	Position	Size	position
C1	0.94	1.216306	1.07	1.66331
C2	0.95	1.097315	0.25	1.02
C3	0.63	0.45607	0.48	1.17047

**Table 3.** Identification results based on the third fitness function (Eq. 12).

Fitness Function $F_3$	ANN Error (mm)		RBF Error (mm)	
	Size	Position	Size	position
C1	0.11	0.494975	0.78	1.203703
C2	0.19	0.221359	0.21	0.408534
C3	0.05	0.284253	0.21	0.550818

The crack identification error varies between 0.22 mm and 1.6 mm in terms of position, and between 0.05 and 1.07 mm in terms of crack size. The results suggest that, overall, the third fitness function corresponds to the highest identification accuracy for both crack size and location. Figure 8 Shows this advantage in crack size prediction. It also shows that Deep ANN is favourable in most cases compared to RBF modelling. This is due to the higher boundary displacement accuracy. However, the fitness evaluation using Deep ANN is very costly compared to RBF. Figure 9 depicts the difference between

Deep ANN and RBF in the computational cost of a single identification run using the YUKI algorithm, with a population of 20 and the maximum number of iterations is 100.

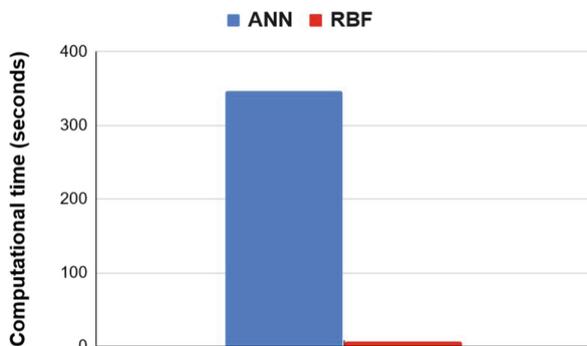


Fig. 9. Computation cost using Deep ANN and RBF.

## 8 Conclusion

In this paper, damage identification in steel plates is presented, using the inverse problem approach. The approach uses the boundary displacement response measured in the presence of an inner crack with different sizes and positions. The finite element method is used to simulate these boundary displacement responses. The inverse problem requires an iterative search with thousands of response evaluations, crack identification researchers investigate alternative solutions to FEM simulation, due to the very expensive computational cost. As well as suggest better-performing search algorithms that require a lower number of iterations.

The performances of Deep ANN and RBF networks for the inverse problem evaluation have been investigated in this work. Using the newly suggested YUKI algorithm. We found that Deep ANN is favourably placed to simulate the boundary displacement response, as it has good stability to unknown crack parameters compared to the RBF method, providing a more consistent prediction error. However, the computation time for RBF is significantly better, which may justify the use of this method.

Boundaries most close to damage are subjected to higher displacement than farther away boundaries. The use of the whole boundary displacement sensors is not favourable for the objective function, because that does not consider the above-mentioned fact, we investigated the segmentation of the boundaries and use cost functions that take into consideration the regional effect of the cracks, and found that the best results are found when considering the average error in all segment plus error in the segment corresponding to the minimum error. This approach will amplify the error in the wrong output, which helps reduce the ill-Posedness of this problem.

## References

1. Amoura, N., Kebir, H., Benzerdjeb, A.: 3D crack identification using the Nelder-Mead Simplex algorithm combined with a random generation of crack positions. *Frattura e Integrità Strutturale* **16**(59), 243–255 (2022)
2. Amoura, N., Kebir, H., Rechak, S., Roelandt, J.: Axisymmetric and two-dimensional crack identification using boundary elements and coupled quasi-random downhill simplex algorithms. *Eng. Anal. Boundary Elem.* **34**, 611–618 (2010)
3. Al Thobiani, F., Khatir, S., Benaissa, B., Ghandourah, E., Mirjalili, S., Wahab, M. A.: A hybrid PSO and Grey Wolf Optimization algorithm for static and dynamic crack identification. *Theoret. Appl. Fract. Mech.* **118**, 103213 (2022)
4. Benaissa, B., Köppen, M., Wahab, M.A., Khatir, S.: Application of proper orthogonal decomposition and radial basis functions for crack size estimation using particle swarm optimization. *J. Phys. Conf. Ser.* **842**(1), 012014 (2017)
5. Khatir, S., Abdel Wahab, M., Tiachacht, S., Le Thanh, C., Capozucca, R., Magagnini, E., et al.: Damage identification in steel plate using FRF and inverse analysis. *Frattura ed Integrità Strutturale-Fracture and Structural Integrity* **58**, 416–433 (2021)
6. Khatir, S., Tiachacht, S., Benaissa, B., Le Thanh, C., Capozucca, R., Abdel Wahab, M.: Damage identification in frame structure based on inverse analysis. In: *Proceedings of the 2nd International Conference on Structural Damage Modelling and Assessment*, pp. 197–211 (2022)
7. Khatir, S., Wahab, M.A., Benaissa, B., Köppen, M.: Crack identification using eXtended IsoGeometric analysis and particle swarm optimization. In: *Fracture, Fatigue and Wear*, pp. 210–222 (2018)
8. Behtani, A., Tiachacht, S., Khatir, T., Khatir, S., Wahab, M.A., Benaissa, B.: Residual Force Method for damage identification in a laminated composite plate with different boundary conditions. *Frattura ed Integrità Strutturale* **16**, 35–48 (2022)
9. Amoura, N., Kebir, H., Rechak, S., Roelandt, J.: Numerical simulation of the behaviour of cracks in axisymmetric structures by the dual boundary element method. In: *Damage and Fracture Mechanics*, ed: Springer, pp. 435–443 (2009)
10. Samir, K., Idir, B., Serra, R., Brahim, B., Aicha, A.: Genetic algorithm based objective functions comparative study for damage detection and localization in beam structures. In: *Journal of Physics: Conference Series*, p. 012035 (2015)
11. Samir, K., Brahim, B., Capozucca, R., Wahab, M.A.: Damage detection in CFRP composite beams based on vibration analysis using proper orthogonal decomposition method with radial basis functions and cuckoo search algorithm. *Compos. Struct.* **187**, 344–353 (2018)
12. Benaissa, B., Belaidi, I., Hamrani, A.: Identifying defect size in two dimensional plates based on boundary measurements using reduced model and genetic algorithm. *J. Sci. Technol.* **2**, 7–12 (2017)
13. Gomes, G.F., Mendez, Y.A.D., da Silva Lopes Alexandrino, P., da Cunha, S.S., Ancelotti, A.C.: A review of vibration based inverse methods for damage detection and identification in mechanical structures using optimization algorithms and ANN. *Arch. Comput. Methods Eng.* **26**, 883–897 (2019)
14. Barai, S., Pandey, P.: Vibration signature analysis using artificial neural networks. *J. Comput. Civ. Eng.* **9**, 259–265 (1995)
15. Zang, C., Imregun, M.: Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection. *J. Sound Vib.* **242**, 813–827 (2001)
16. Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., Inman, D.J.: Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **388**, 154–170 (2017)

17. Stavroulakis, G.E., Engelhardt, M., Likas, A., Gallego, R., Antes, H.: Neural network assisted crack and flaw identification in transient dynamics. *J. Theor. Appl. Mech.* **42**, 629–649 (2004)
18. Stavroulakis, G., Antes, H.: Neural crack identification in steady state elastodynamics. *Comput. Methods Appl. Mech. Eng.* **165**, 129–146 (1998)
19. Khaleghi, M., Haghighat, E., Vahab, M., Shahbodagh, B., Khalili, N.: Fracture characterization from noisy displacement data using artificial neural networks. *Eng. Fract. Mech.* **271**, 108649 (2022)
20. Benaissa, B., Hocine, N.A., Khatir, S., Riahi, M.K., Mirjalili, S.: YUKI algorithm and POD-RBF for Elastostatic and dynamic crack identification. *J. Comput. Sci.* **55**(1851), 101451 (2021)
21. Benaissa, B., Ait Hocine, N., Belaidi, I., Hamrani, A., Pettarin, V.: Crack identification using model reduction based on proper orthogonal decomposition coupled with radial basis functions. *Struct. Multidiscip. Optim.* **54**(2), 265–274 (2016). <https://doi.org/10.1007/s00158-016-1400-y>
22. Stavroulakis, G., Antes, H.: Nondestructive elastostatic identification of unilateral cracks through BEM and neural networks. *Comput. Mech.* **20**, 439–451 (1997)
23. Buljak, V., Maier, G.: Proper orthogonal decomposition and radial basis functions in material characterization based on instrumented indentation. *Eng. Struct.* **33**, 492–501 (2011)
24. Arora, Y., Singhal, A., Bansal, A.: A study of applications of RBF network. *Int. J. Comput. Appl.* **94**(2), 17–20 (2014)
25. Goodfellow, I., Bengio, Y., Courville, A.: *Deep learning*. MIT press, Cambridge (2016)
26. LeCun, Y., Bengio, Y., Hinton, G.: *Deep learning*. *nature*, **521**, 436–444 (2015)
27. Hinton, G.E.: Learning multiple layers of representation. *Trends Cogn. Sci.* **11**, 428–434 (2007)
28. Sharma, O.: Deep challenges associated with deep learning. In: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 72–75 (2019)