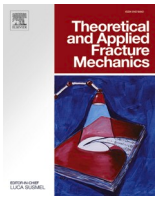
Theoretical and Applied Fracture Mechanics 134 (2024) 104761

Contents lists available at ScienceDirect

Theoretical and Applied Fracture Mechanics

journal homepage: www.elsevier.com/locate/tafmec



Interpolating CTS specimens' mode I and II stress intensity factors using artificial neural networks

R. Baptista [a,b], V. Infante [b], L.F.P. Borrego [c], E.R. Sérgio [d], D.M. Neto [d], F.V. Antunes [d,*]

[a] UnIRE, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, R. Conselheiro Emídio Navarro 1, 1959-007 Lisboa, Portugal
[b] LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
[c] Polytechnic University of Coimbra, Rua da Misericórdia, Lagar dos Cortiços, S. Martinho do Bispo, 3045-093 Coimbra, Portugal
[d] Univ Coimbra, Centre for Mechanical Engineering, Materials and Processes (CEMMPRE), Department of Mechanical Engineering, Portugal

ARTICLE INFO

Keywords:
CTS specimen
Mixed Mode Loading
Stress intensity factor
Artificial Neural Networks

ABSTRACT

Fracture mechanics parameters, such as the stress intensity factor (SIF), are fundamental for the analysis of fracture, fatigue crack growth and crack paths. SIFs of a cracked body can be determined either experimentally or numerically. Analytical solutions of SIF are very useful, but their determination from discrete values can be extremely complex when there are many independent variables. In this paper, artificial neural networks (ANN) are proposed to predict mode I and II stress intensity factors in a CTS specimen under mixed mode loading conditions. Trained with numerical data, the performance of different network architectures and back-propagation algorithms was assessed. Using at least 10 neurons, in the hidden layers, made it possible for the designed solution to match the performance of analytical solutions. Increasing the number of neurons, allowed the model performance to improve up to 90%, when compared with previous analytical solutions. This increases the quality of fracture and fatigue studies done with the CTS sample.

1. Introduction

The stress intensity factor (SIF), K, quantifies the level of stress singularity ahead of a crack in materials with elastic behaviour. This parameter has been widely used in fracture and fatigue studies of different materials, metallic and non-metallic [1]. It is particularly adequate for brittle materials, but also for ductile materials loaded within the small-scale yielding (SSY) regime. The brittle failure is supposed to occur when K reaches the fracture toughness of the material (K_{Ic}). Fatigue crack growth (FCG) rate is usually plotted versus the SIF range, ΔK. Different issues of FCG have been solved using the concepts of crack closure [2], and two-parameter approaches based on K_{max} and ΔK [3–5].

The SIF is usually given by:

$$K = Y\sigma\sqrt{\pi a} \tag{1}$$

where Y is the geometric factor, σ the remote stress and a the crack length. The geometric factor depends on several parameters, namely the geometry of the cracked component, the geometry of the crack (interior, surface, through-thickness), the position of the crack, the crack length, the crack shape, and the loading. It is usually calculated using the finite element method (FEM). Direct and indirect methods use the outputs of the FEM, i.e., the displacement, strain and stress fields around the crack tip, to obtain K [6–8]. Recently Triclot *et al.* [9] conducted a study over the influence of different boundary conditions, regarding the numerical SIFs determination. Experimental approaches have also been used, namely those based on Digital Image Correlation (DIC). An over-deterministic approach was proposed [10,11], considering analytical models of crack tip fields [12]. Experimental results are influenced by the chosen collection zone and noise to signal ratio [13].

Many analytical solutions of K can be found in literature for different cracked geometries. ASTM E647 presents solutions for standard CT and MT specimens [14]. Tada *et al.* [15], Murakami [16] and Carpinteri [17] present compilations of K solutions. However, in some cases the fitting of analytical solutions involves a significant number of parameters and is not a straightforward process. Alegre *et al.* [18] proposed a solution for an embedded elliptical crack in a round bar subjected to tensile load. The crack position in the cross-section and the aspect ratio were characterised by three parameters, the two semi-axes of the ellipse (a and c) and the width of the ligament (h). The bar is defined by the radius (R), and the tensile load applied at the ends of the bar is defined by the constant uniaxial stress (σ_0). Antunes *et al.* [19] proposed K solutions for

* Corresponding author.
E-mail address: fernando.ventura@dem.uc.pt (F.V. Antunes).

https://doi.org/10.1016/j.tafmec.2024.104761
Received 28 September 2024; Received in revised form 7 November 2024; Accepted 14 November 2024
Available online 19 November 2024

Nomenclature

AI	Artificial Intelligence
ANN	Artificial Neural Network
BR	Bayesian Regularization backpropagation
CTS	Compact Tension Shear specimen
DL	Deep Learning
FCG	Fatigue Crack Growth
FEM	Finite Element Method
GRNN	Generalized Regression Neural networks
HPO	Hyperparameter optimization
LM	Levenberg-Marquardt backpropagation
ML	Machine Learning
MTS	Maximum tangential stress
SIF	Stress Intensity Factor
SR	Symbolic Regression
SSY	Small-scale Yielding
a	crack length
α	loading direction
B	Specimen's thickness
β	slope of crack flanks at crack tip
F	Applied load
ΔK	Range of stress intensity factor
K	Stress intensity factor
K_{Ic}	Fracture toughness
K_{max}	Maximum stress intensity factor
x	Crack tip x coordinate
y	Crack tip y coordinate
Y	Crack geometric factor
W	Specimen's width

corner cracks with quarter-circular shape and with tunnelling effect. Newman [20,21] proposed a set of equations for corner and surface cracks.

The compact tension shear (CTS) specimen has been widely used for mixed-mode loading (I/II) studies of fatigue and fracture. An apparatus, that can be used in conventional universal testing machines (visible in Fig. 2), is used to rotate the specimen applying different mixed mode load ratios. However, this implies that modes I and II are always in-phase. The fact that a multi-axial machine is not necessary, is one of the main reasons for the wide use of this type of specimen. In fact, the CTS specimen has been used to study not only homogeneous metallic materials [22,23], but also aluminium-epoxy interface [24], glass fibers/epoxy composites [25], orthotropic materials [26] and bonded joints [27]. The K solution proposed by Sander and Richard is widely used [28]:

$$K_I = \frac{F}{WB}\sqrt{\pi a}\,\frac{\cos\alpha}{1-\frac{a}{w}}\sqrt{\frac{0.26+2.65\frac{a}{w-a}}{1+0.55\frac{a}{w-a}-0.08\left(\frac{a}{w-a}\right)^2}} \tag{2}$$

$$K_{II} = \frac{F}{WB}\sqrt{\pi a}\,\frac{\sin\alpha}{1-\frac{a}{w}}\sqrt{\frac{-0.23+1.40\frac{a}{w-a}}{1-0.67\frac{a}{w-a}+2.08\left(\frac{a}{w-a}\right)^2}} \tag{3}$$

where F is the applied load. W and B are the width and the thickness of the CTS specimen, respectively, α is the loading angle and a is the crack length. However, these equations are only valid in the range of a/W between 0.55 and 0.7 and assume that the crack is perpendicular to the longitudinal axis of the specimen. Therefore, they are only valid for stationary cracks, i.e., for fracture studies. Antunes *et al.* [29] extended these solutions with the inclusion of the angle at the crack tip.

Alternative strategies are required to obtain the field of stress intensity factor from discrete values in wide parametric spaces. The shape functions of the FEM can be used in this context. These functions are used to interpolate displacements inside the finite elements from nodal values. Therefore, provided that a grid of points is defined in the parametric space, and that the K values are obtained at these points, this strategy can be used to interpolate values. More recently, artificial intelligence (AI) tools, have been proposed, but are still considered data-driven, black-box solutions to this day [30]. They are normally used as machine learning (ML) tools, with the ability to learn or capture data behaviour without the need for a previously known physical or theoretical model. AI methods have been used in fracture mechanics problems for some time now, but many authors still believe there are numerous issues and opportunities for improvement [31]. They have been successfully used on fracture [32], fatigue crack propagation [33,34] and crack detection [31,35], using tools, like artificial neural networks (ANN), random forests, or support vector machines.

In the field of FCG, Wang *et al.* [36] used generalized regression neural networks (GRNN) to predict crack paths in CT specimens and plates with two holes. GRNN have unique solutions and can be quickly trained using both experimental and numerical data. Unfortunately, these ML solutions cannot be tailored, therefore authors like Baptista *et al.* [34,37] used both shallow and deep learning (DL) solutions to design networks with performance optimized for the type of data used to train them. In fact, Gope *et al.* [38] considered the selection of the input and output variables the most important parameter in network design. They were able to accurately predict crack propagation angles in notch plates, using ANN and the maximum tangential stress (MTS) criterion. Fatigue life prediction applications have also been explored by many authors. Pujol *et al.* [39] and Barbosa *et al.* [40] used ANN, trained with experimental data, to predict specimen fatigue life under variable amplitude loading with different mean stress. Since there is no need for a supporting theoretical background, fatigue life is predicted based on the nature of the data, without the constraints of a physical or empirical model.

Fracture analysis can also be studied using AI tools. Bahrami *et al.* [32] used ANN trained with different backpropagation algorithms to predict CTS specimens' fracture loads under mixed mode conditions. According to these authors, choosing the right training algorithm is fundamental to achieve the best network performance, but the nature of the problem inputs and outputs can affect the training algorithm result.

There are still several challenges to be solve in this area. The network architecture is also an important parameter to select. Zhao *et al.* [41] used AI tool to predict both fatigue lives and crack trajectories in CTS specimens. While fatigue life was accurately predicted using long short-term memory networks, with neuron feedback, crack paths were better predicted using feedforward networks, where information only travels in one direction. Once the network architecture is selected, its performance can be affected by underfitting, overfitting, or instability. Underfitting occurs when the network is not able to accurately capture the system behaviour, resulting in inadequate model performance. Overfitting occurs when the excessive number of network parameters or training epochs leads to over-adaptation to noise and random variations in the training dataset, with the network failing to generalize their performance on new datasets. Finally, instability occurs when the training dataset is not large enough to provide the necessary information for network training, causing model parameters to change each time the network is trained [42]. According to Liang *et al.* [42], many of these problems can be solved by adding more points to the training dataset. If these are not available, methodologies as the Delaunay interpolation can be used to provide data augmentation and to expand the training dataset.

As ANN architectures become more complex, many different

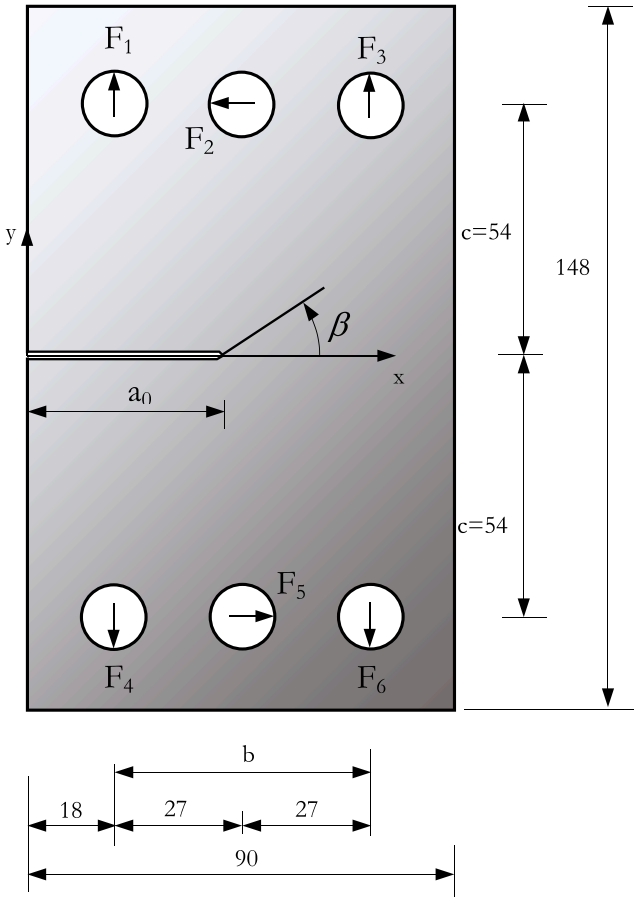


Fig. 1. Geometry and loading of CTS specimen.

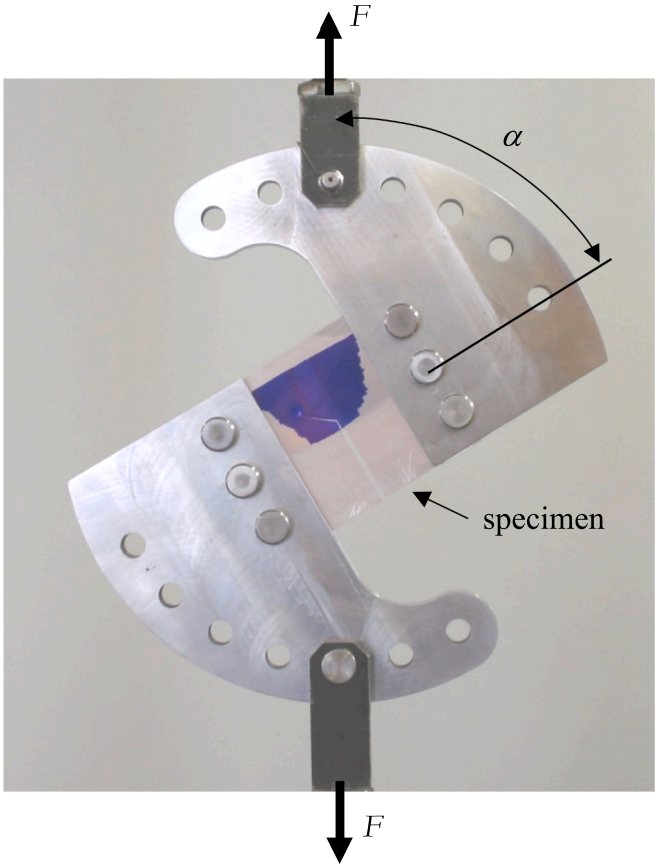


Fig. 2. Mixed-mode loading device (α = 60°).

hyperparameters become available to tune network performance. Recently, different authors have turned their attention into neural network hyperparameter optimization (HPO) [43]. While shallow and deep neural network main hyperparameters include the number of hidden layers, number of neurons per layer, activation function and backpropagation algorithm, convolutional neural networks hyperparameters also include the kernel size, stride, padding, and dropout and learning rate. According to Iliadis *et al.* and Mamun *et al.*, HPO algorithms can be divided into: Grid search and random search methods, sequential model-based optimization, *meta*-heuristic optimization, numerical approach-based optimization (grid search) and statistical modelling-based optimization [43,44]. Grid search is a very simple optimization procedure to setup, but as a brute-force methodology has a very high computational cost. Sequential models, like the Tree-structured Parzen Estimator [45], are based on the Bayesian approach and use random forests as surrogate models to process categorical hyperparameters in very large search spaces. These methodologies can scale better than grid search as the problem dimension increases. Meta-heuristic algorithms are based in bio-inspired methodologies, mimicking animal behaviour, music theory or genetic evolution showing improved training efficacy over grid search [46]. Numerical approach methodologies use direct search methods to search for the optimal solution without using derivatives. Finally, statistical modelling expands and structures the grid and random search methods using a systematic approach based on the Bayes' theorem applied to the hyperparameter map [44].

While the previously described techniques can be effective, they may lack transparency and interpretability. They are considered black-box approaches, providing efficient tools, even when no theoretical data model is available [47]. Using AI, a new set of tools has recently emerged in the field of data-driven modelling. Symbolic regression (SR) is a ML tool that allows for an intelligent search of the optimal closed-form mathematical equation [48]. In fact, different authors have used this approach to solve facture mechanics problems. Melching *et al.* used SR and numerical data to train a model able to correct the crack tip location using the experimental stress distribution around the crack tip, expanding the previous work from Rethoré [47]. Merrell *et al.* used SR to develop new SIF solutions for elliptical cracks, providing simpler solutions with the same performance of the Raju-Newman equations, or more complex, but more effective new solutions for this problem [49].

The objective here is to test the applicability of AI to obtain SIF values for the CTS specimen with improved accuracy. The AI tool was trained based on previously obtained FEM results. Antunes *et al.* [29] developed an extensive numerical work, and two analytical solutions were fitted for modes I and II SIFs. However, the analytical solutions are relatively complex and does not adequately fit the numerical results in some regions of the parametric space. A better solution is therefore searched here using AI. Both shallow and DL solutions were trained with the previously determined SIF solutions. The effect of the network architecture was assessed, as well as their final performance. The best solutions were compared with the previous analytical models.

2. Analytical K solution for the CTS specimen

The CTS specimen [50] has been widely used to study fracture and fatigue under in-phase mixed I/II loading conditions [22–28]. The geometry of the specimen is defined in Fig. 1. It has a width of 90 mm, and a length of 148 mm. There are three holes at each end, which are used for load application. As indicated in Fig. 2, a uniaxial tension testing machine is used to apply the load, and the change of loading mode is achieved by rotating the specimen. The loading angle, α, is defined between the longitudinal axis of the specimen and the load direction. The

Table 1
ANN input variables and corresponding region of interest.

Variable	Lower bound	Increment	Upper bound
x[mm]	32.5	5	67
y[mm]	0	5	15
β[deg.]	0	10	40
α[deg.]	0	10	60

Table 2
ANN architectures with different hidden layers and number of neurons parameters.

One hidden layer	# neurons	Two hidden layers	# neurons	Three hidden layers	# neurons
(5)	5	(5,5)	10	(5,5,5)	15
(10)	10	(5,10)	15	(5,5,10)	20
(15)	15	(10,5)	15	(5,10,5)	20
(20)	20	(10,10)	20	(10,5,5)	20
(25)	25	(10,15)	25	(5,10,10)	25
(30)	30	(15,10)	25	(10,5,10)	25
(45)	45	(5,15)	20	(10,10,5)	25
(60)	60	(15,5)	20	(10,10,10)	30
		(20,20)	40	(10,10,15)	35
		(25,25)	50	(10,15,10)	35
				(15,10,10)	35
				(15,15,15)	45

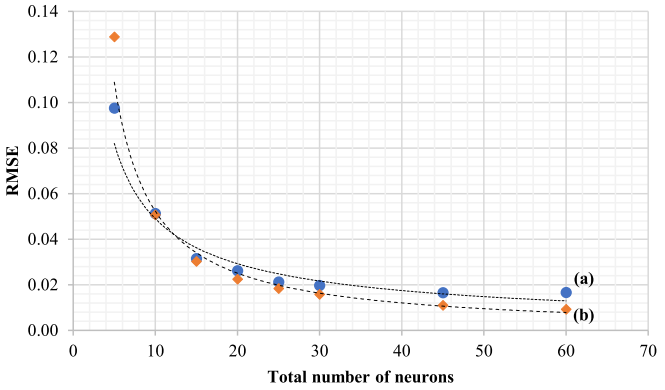


Fig. 3. Shallow learning (one hidden layer) Networks' RMSE, trained using (a) LM and (b) BR algorithm.

main geometrical parameters are the coordinates of the crack tip (x, y), the slope of crack flanks at crack tip (β), specimen's width (W) and thickness (B). Additionally, the stress intensity factors depend on applied force (F) and loading direction (α).

An extensive set of numerical simulations was previously developed [29], involving a total number of 1120 cracked geometries. 24 different crack tip locations were considered, defined by x and y coordinates of the crack tip. Additionally, seven values were considered for the loading angle, α (0, 10, 20, 30, 40, 50, 60°), while five values were considered for β (0, 10, 20, 30 and 40°). J-integral was used to calculate K_I and K_{II}:

$$K_I, K_{II} = \widetilde{f}(\sigma, W, x, y, \alpha, \beta) \tag{4}$$

where the stress σ is given by $F/(Wt)$. The number of independent variables was reduced using Buckingham's theorem of non-dimensional analysis [51]. Considering σ and x as the primary variables, the following non-dimensional relations were obtained:

$$Y_I = \frac{K_I}{\sigma.\sqrt{\pi.x}} = \widetilde{f}(\frac{x}{w}, \frac{y}{w}, \alpha-\beta, \beta) \tag{5}$$

$$Y_{II} = \frac{K_{II}}{\sigma.\sqrt{\pi.x}} = \widetilde{f}(\frac{x}{w}, \frac{y}{w}, \alpha-\beta, \beta) \tag{6}$$

An analytical solution with 39 constants was fitted to the FEM results:

$$Y_I = m(W/(W-X))^{1.5} + b \tag{7}$$

$$m = a_m.(a-b)^2 + b_m.(a-b) + c_m$$

$$a_m = m_{am}.(y/w) + b_{am}$$

$$m_{am} = 8.4471\times10^{-9}.\beta^2 - 4.6826\times10^{-8}.\beta - 1.5894\times10^{-4}$$

$$b_{am} = 1.2076\times10^{-8}\beta^2 - 1.9219\times10^{-7}.\beta - 1.3955\times10^{-4}$$

$$b_m = m_{bm}.(y/w) + b_{bm}$$

$$m_{bm} = -2.3457\times10^{-6}\beta^2 - 2.9335\times10^{-4}\beta + 4.1625\times10^{-2}$$

$$b_{bm} = 9.6990\times10^{-7}\beta^2 - 2.4575\times10^{-4}\beta - 2.8677\times10^{-4}$$

$$c_m = m_{cm}.(y/w) + b_{cm}$$

$$m_{cm} = -2.7508\times10^{-4}\beta^2 + 4.3628\times10^{-2}\beta - 3.7856\times10^{-2}$$

$$b_{cm} = -1.5507\times10^{-4}\beta^2 - 9.5592\times10^{-4}\beta + 1.0744$$

$$b = a_b.(\alpha-\beta)^2 + b_b.(\alpha-\beta) + c_b$$

$$a_b = a_{ab}.\beta^2 + b_{ab}.\beta + c_{ab}$$

$$a_{ab} = 4.9864\times10^{-8}.(y/w) + 1.4723\times10^{-8}$$

$$b_{ab} = -2.9412\times10^{-6}.(y/w) - 1.9886\times10^{-6}$$

$$c_{ab} = -2.6726\times10^{-4}.(y/w) + 3.0499\times10^{-5}$$

$$b_b = a_{bb}.\beta^2 + b_{bb}.\beta + c_{bb}$$

$$a_{bb} = -4.3391\times10^{-5}.(y/w)^2 + 2.1155\times10^{-6}.(y/w) - 5.3868\times10^{-6}$$

$$b_{bb} = 2.1858\times10^{-3}.(y/w)^2 - 9.6565\times10^{-4}.(y/w) + 5.8686\times10^{-4}$$

$$c_{bb} = 5.7786\times10^{-2}.(y/w) - 2.6952E-03$$

$$c_b = a_{cb}.\beta^2 + b_{cb}.\beta + c_{cb}$$

$$a_{cb} = -6.9194\times10^{-4}.(y/w) + 2.6846E-04$$

$$b_{cb} = 7.8256\times10^{-2}.(y/w) + 5.9363E-04$$

$$c_{cb} == 2.0543.(y/w)^2 - 6.2440\times10^{-3}.(y/w) - 1.5349\times10^{-1}$$

The solution proposed for Y_I fitted reasonably well the FEM results, with average and maximum differences of 0.53 % and 1.9 %, respectively. Similarly, an analytical solution with 54 fitting parameters was proposed for mode II of loading:

$$Y_{II} = a.sin(|\alpha-\beta| - \tau) \tag{8}$$

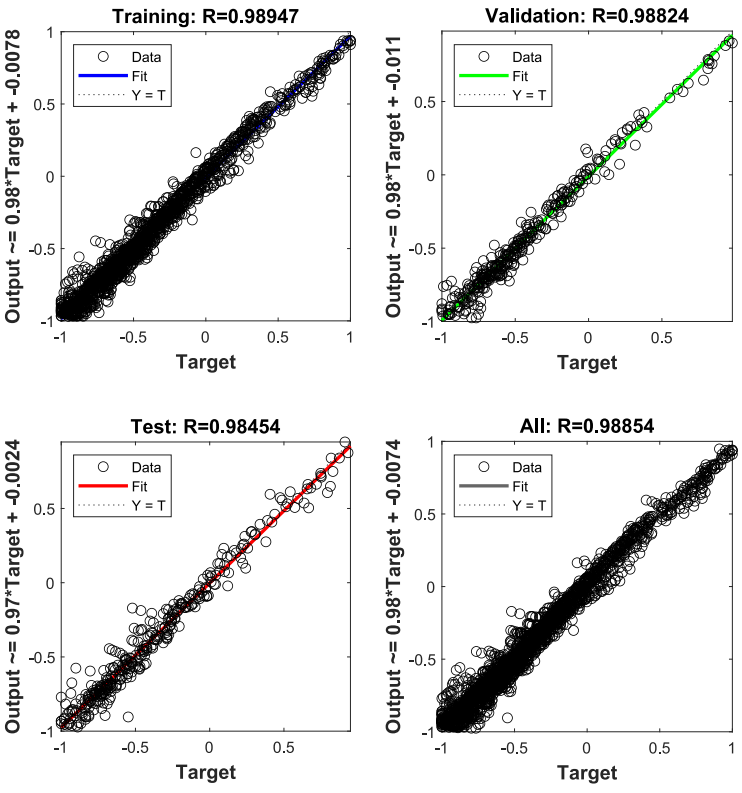
$$a = a_a.(x/w)^2 + b_a.(x/w) + c_a$$

$$a_a = a_{aa}.\beta^2 + b_{aa}.\beta + c_{aa}$$

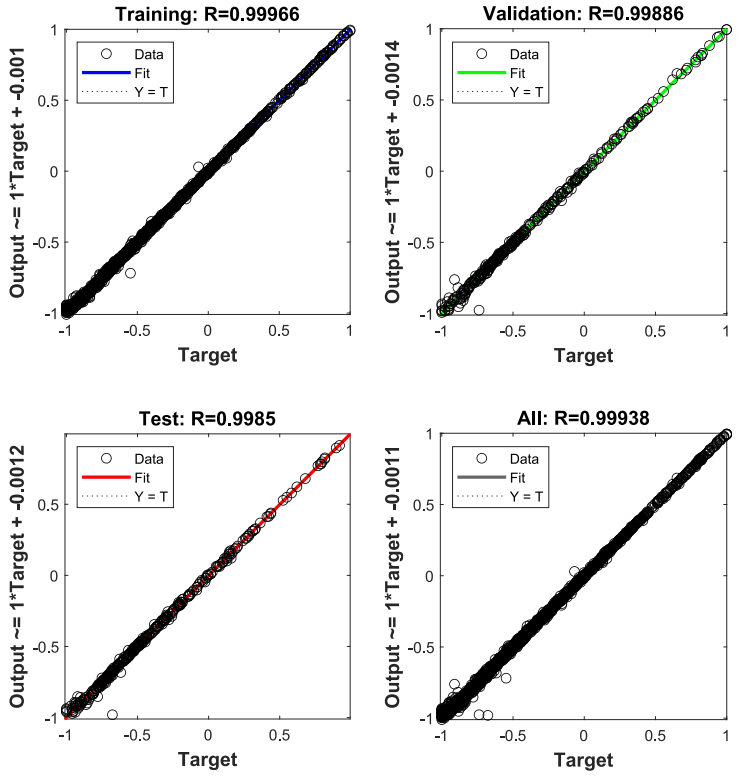
$$a_{aa} = -1709.472357.(y/w)^2 + 519.545852(y/w) + 21.384622$$

$$b_{aa} = 959.716917.(y/w)^2 - 249.236159(y/w) + 6.561228$$

$$c_{aa} = -98.482311.(y/w)^2 + 4.234582(y/w) + 1.006272$$



(a)



(b)

Fig. 4. MATLAB's regression (a) LM_5 (5 neurons) (b) LM_45 (45 neurons). (one hidden layer).

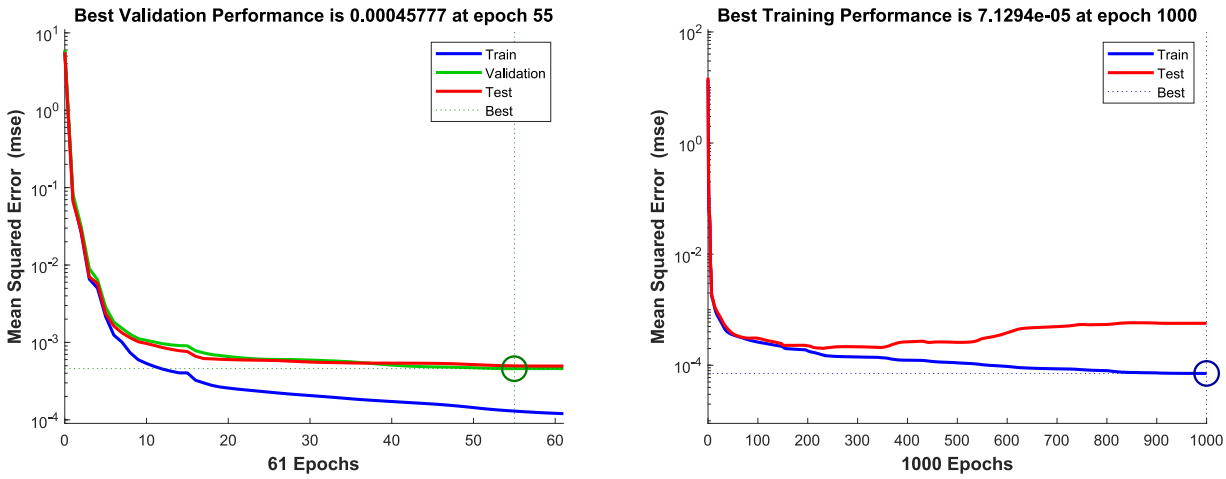


Fig. 5. MATLAB's (a) LM_45 (b) BR_45. (45 neurons).

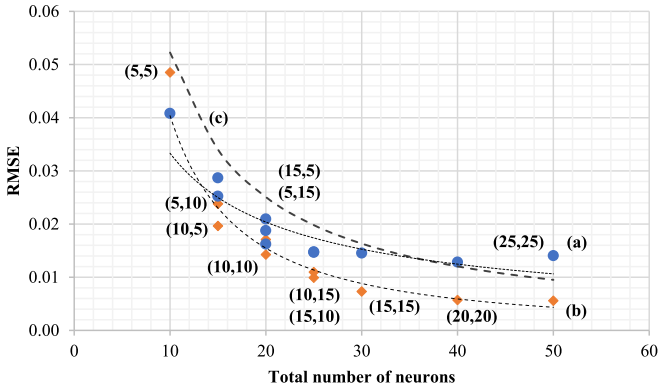


Fig. 6. Machine learning (two hidden layers) Networks' RMSE, trained using (a) LM and (b) BR algorithm. For comparison (c) the shallow learning model trained with the BR algorithm trendline has been added.

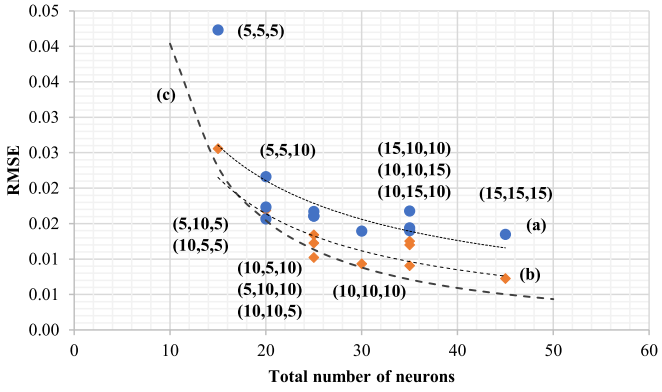


Fig. 7. Deep learning (three hidden layers) Networks' RMSE, trained using (a) LM and (b) BR algorithm. For comparison (c) the shallow two hidden layers model trained with the BR algorithm trendline has been added.

$$b_a = a_{ba}.\beta^2 + b_{ba}.\beta + c_{ba}$$

$$a_{ba} = 1953.567396.(y/w)^2 - 541.35909(y/w) - 19.747265$$

$$b_{ba} = -1101.79278.(y/w)^2 + 255.808188(y/w) - 7.044048$$

$$c_{ba} = 113.285142.(y/w)^2 - 3.803387(y/w) + 0.704463$$

$$c_a = a_{ca}.\beta^2 + b_{ca}.\beta + c_{ca}$$

$$a_{ca} = -529.091514.(y/w)^2 + 143.768493(y/w) + 4.296419$$

$$b_{ca} = 299.064879.(y/w)^2 - 71.861075(y/w) + 1.74509$$

$$c_{ca} = -27.688878.(y/w)^2 + 1.127385.(y/w) + 0.773005$$

$$t = a_t.(x/w)^2 + b_t.(x/w) + c_t$$

$$a_t = a_{at}.\beta^2 + b_{at}.\beta + c_{at}$$

$$a_{at} = -31.328451.(y/w)^2 - 189.201651(y/w) - 13.342750$$

$$b_{at} = -75.128472.(y/w)^2 + 152.527046(y/w) + 11.48976$$

$$c_{at} = 32.715576.(y/w)^2 - 17.576125(y/w) - 0.138782$$

$$b_t = a_{bt}.\beta^2 + b_{bt}.\beta + c_{bt}$$

$$a_{bt} = -15.361974.(y/w)^2 + 209.112802(y/w) + 11.726069$$

$$b_{bt} = 116.101593.(y/w)^2 - 166.317934(y/w) - 9.360830$$

$$c_{bt} = -41.982624.(y/w)^2 + 19.603112(y/w) + 0.083306$$

$$c_t = a_{ct}.\beta^2 + b_{ct}.\beta + c_{ct}$$

$$a_{ca} = 31.329018.(y/w)^2 - 53.553350(y/w) - 2.292627$$

$$b_{ca} = -50.034429.(y/w)^2 + 44.725843(y/w) + 1.819130$$

$$c_{ct} = 12.967452.(y/w)^2 - 5.665219(y/w) - 0.011360$$

The units of β, α and τ are radians. The parametric region where this solution is valid is: α ∈ [0, 60°]; x/w ∈ [0.4, 0.75 mm]; y/w ∈ [0, 0.167]. This solution had an average difference of 10.8 % relatively to the

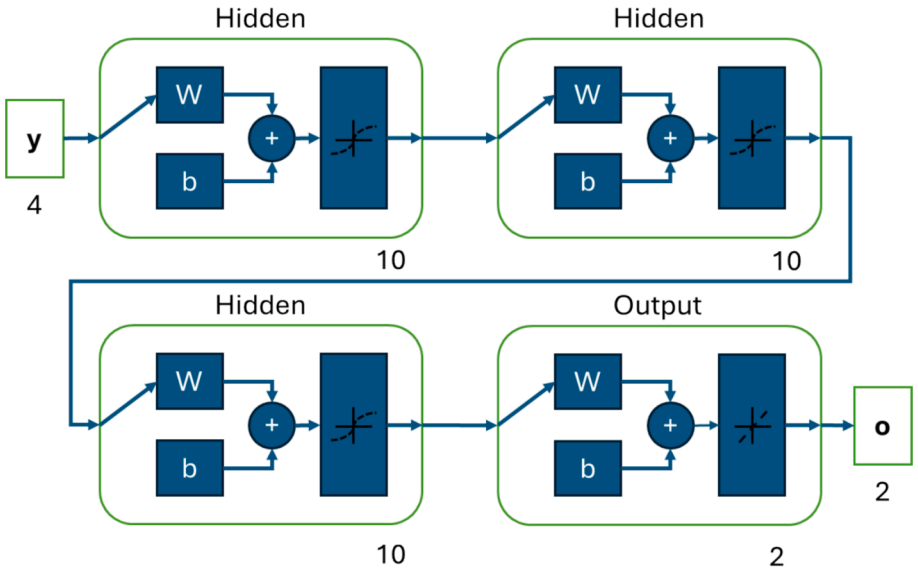


Fig. 8. (10,10,10)_LM network structure with 4 inputs, 30 neurons (10 per hidden layer) and 2 outputs.

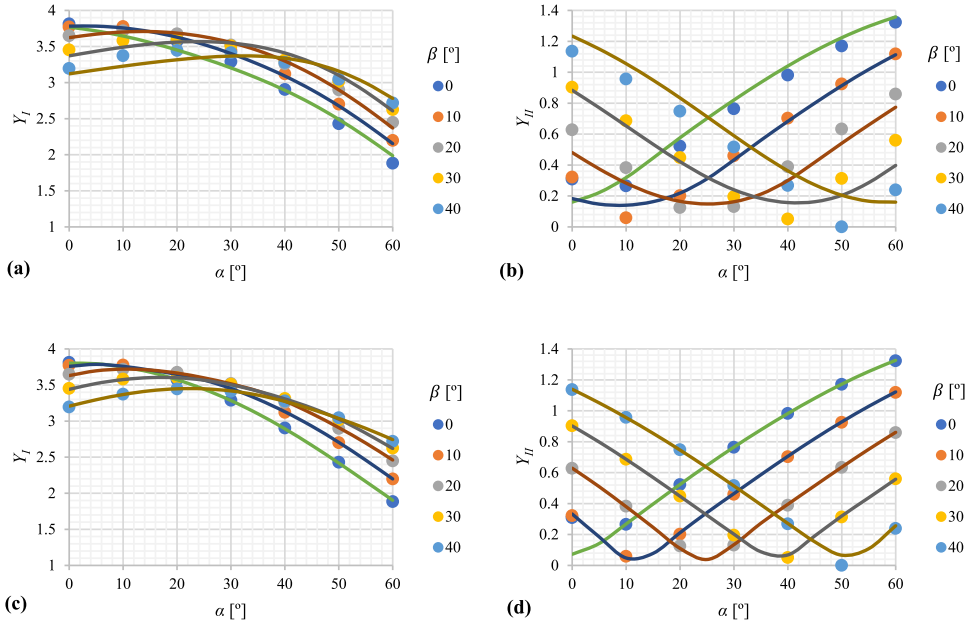


Fig. 9. Non-dimensional SIFs (a) (b) Y_I and Y_{II} variation with α for a (5)_LM network. (c) (d) Y_I and Y_{II} variation with α for a (10,10,10)_LM network. (x = 52.5 mm, y = 0 mm).

numerical values, showing very large differences in some regions of the parametric space. This average was, once again, calculated using the absolute values of the differences. These solutions of Y_I and Y_{II} were verified by Jin *et al.* [52], in a more limited approach. They determined the SIF evolution with the loading angle α for slanted cracks always lying on the CTS specimen centre. Similar results were obtained, although the crack slope considered by these authors was inversely proportional to the loading angle. This minimizes Y_{II} making the mode mixity ≈ 1 [52]. These authors also mentioned that SIF are mainly dependent on the loading angle, and less so on the specimen thickness. Besides, once crack propagation has begun, Y_{II} will vanish rapidly, while Y_I will increase with the crack length, as the crack propagates in mode I. Finally, Shlyannikov *et al.* [53,54] have shown how plasticity influences SIFs, with a stronger influence under plane stress conditions.

3. AI model

An alternative approach to these analytical solutions is proposed here, based on AI tools. A data driven approach relies on data to determine the relationship between selected input and output variables, therefore, it does not require a previously defined theoretical or physics-based model. ML models, including the more recent deep learning AI tools, are essentially black-boxes that can automatically determine the relationships between influencing factors and targets using a variety of training algorithms [30]. ANN are a specific subset of ML or DL tools, based on the used number of hidden layers. They are structured as fully interconnected networks of nodes or neurons, functioning based on the neurological principles of the human brain [31]. Each input variable is represented by an individual neuron on the input layer, and each neuron is fully connected to the neurons of the first hidden layer. In the case of a DL model, several hidden layers, with different number of neurons, will

(c) (d) Y_I and Y_{II} variation with α for a (10,10,10)_LM network. (x = 52.5 mm, y = 0 mm)

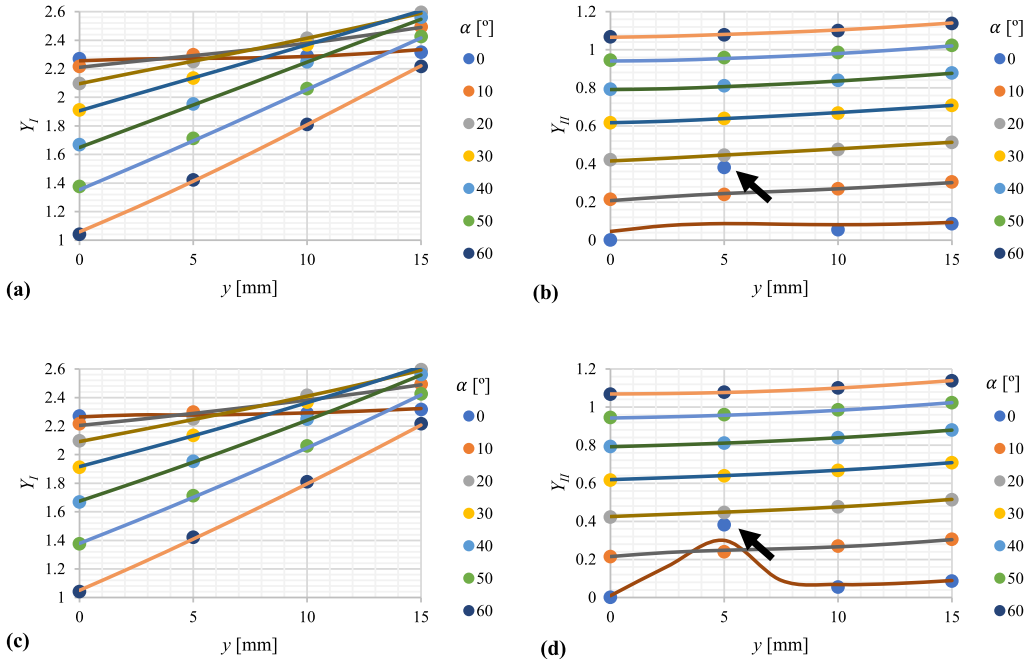


Fig. 10. Non-dimensional SIFs (a) (b) Y_I and Y_{II} variation with y for a (10,10,10)_LM network. (c) (d) Y_I and Y_{II} variation with y for a (10,10,10)_BR network. (x = 37.5 mm, β = 0°; data variation and overfitting to data by RB trained network).

Table 3
Non-dimensional SIFs Y_I variation with α and β for a (5)_LM network (x = 52.5 mm, y = 0 mm). The corresponding analytical value obtained in [29] is shown for comparison in parentheses.

α[°]β[°]	0	10	20	30	40
0	3.77 (3.84)	3.78 (3.8)	3.62 (3.67)	3.37 (3.46)	3.12 (3.21)
10	3.65 (3.76)	3.76 (3.78)	3.7 (3.72)	3.49 (3.57)	3.23 (3.36)
20	3.45 (3.57)	3.63 (3.66)	3.69 (3.66)	3.56 (3.57)	3.32 (3.42)
30	3.2 (3.29)	3.4 (3.43)	3.56 (3.5)	3.55 (3.48)	3.37 (3.38)
40	2.89 (2.91)	3.09 (3.11)	3.3 (3.23)	3.41 (3.27)	3.34 (3.23)
50	2.49 (2.43)	2.68 (2.68)	2.9 (2.86)	3.1 (2.97)	3.16 (2.98)
60	1.99 (1.86)	2.15 (2.15)	2.37 (2.39)	2.61 (2.56)	2.78 (2.64)

Table 4
Non-dimensional SIFs Y_{II} variation with α and β for a (5)_LM network (x = 52.5 mm, y = 0 mm). The corresponding analytical value obtained in [29] is shown for comparison in parentheses.

α[°]β[°]	0	10	20	30	40
0	0.16 (0.02)	0.18 (0.32)	0.48 (0.62)	0.88 (0.9)	1.23 (1.13)
10	0.32 (0.28)	0.14 (0.06)	0.28 (0.38)	0.65 (0.68)	1.06 (0.95)
20	0.58 (0.54)	0.22 (0.21)	0.17 (0.13)	0.42 (0.45)	0.83 (0.74)
30	0.82 (0.78)	0.43 (0.46)	0.16 (0.13)	0.24 (0.19)	0.59 (0.51)
40	1.04 (0.99)	0.68 (0.7)	0.3 (0.39)	0.16 (0.06)	0.36 (0.26)
50	1.22 (1.18)	0.91 (0.92)	0.54 (0.63)	0.2 (0.32)	0.2 (0.01)
60	1.36 (1.33)	1.11 (1.12)	0.77 (0.86)	0.4 (0.56)	0.16 (0.24)

Table 5
Non-dimensional SIFs Y_I variation with α and β for a (10,10,10)_LM network (x = 52.5 mm, y = 0 mm). The corresponding analytical value obtained in [29] is shown for comparison in parenthesis.

α[°]β[°]	0	10	20	30	40
0	3.8 (3.84)	3.76 (3.8)	3.63 (3.67)	3.44 (3.46)	3.21 (3.21)
10	3.75 (3.76)	3.76 (3.78)	3.72 (3.72)	3.58 (3.57)	3.37 (3.36)
20	3.57 (3.57)	3.65 (3.66)	3.67 (3.66)	3.6 (3.57)	3.45 (3.42)
30	3.28 (3.29)	3.45 (3.43)	3.52 (3.5)	3.5 (3.48)	3.42 (3.38)
40	2.89 (2.91)	3.13 (3.11)	3.28 (3.23)	3.31 (3.27)	3.27 (3.23)
50	2.42 (2.43)	2.71 (2.68)	2.91 (2.86)	3.03 (2.97)	3.04 (2.98)
60	1.9 (1.86)	2.2 (2.15)	2.46 (2.39)	2.62 (2.56)	2.74 (2.64)

Table 6
Non-dimensional SIFs Y_{II} variation with α and β for a (10,10,10)_LM network (x = 52.5 mm, y = 0 mm). The corresponding analytical value obtained in [29] is shown for comparison in parentheses.

α[°]β[°]	0	10	20	30	40
0	0.07 (0.02)	0.33 (0.32)	0.63 (0.62)	0.9 (0.9)	1.14 (1.13)
10	0.27 (0.28)	0.05 (0.06)	0.38 (0.38)	0.69 (0.68)	0.96 (0.95)
20	0.52 (0.54)	0.21 (0.21)	0.11 (0.13)	0.45 (0.45)	0.75 (0.74)
30	0.77 (0.78)	0.46 (0.46)	0.13 (0.13)	0.2 (0.19)	0.51 (0.51)
40	0.98 (0.99)	0.71 (0.7)	0.4 (0.39)	0.07 (0.06)	0.27 (0.26)
50	1.17 (1.18)	0.93 (0.92)	0.64 (0.63)	0.32 (0.32)	0.07 (0.01)
60	1.33 (1.33)	1.12 (1.12)	0.86 (0.86)	0.56 (0.56)	0.26 (0.24)

be sequentially connected to the final output layer. On the other hand, shallow ML model will be comprised of a single hidden layer. The information travels in one direction only (feedforward network), connecting neurons using a specific weight (W^l) matrix and bias (b^l) vectors

$$a^l = f^l\left(W^l \cdot a^{l-1} + b^l\right) \tag{9}$$

where a^l is the output vector of layer l, and a^{l-1} its input vector (the output from layer $l-1$). f^l represents the transfer function between layers, usually mapping the outputs into the [-1,1] domain. Transfer functions can be sigmoidal or linear. MATLAB ANN implementation uses sigmoidal functions between hidden layers and a linear function to map the networks' outputs. According to Yonaba *et al.* [55] using a sigmoidal function between hidden layers improves the network performance, even for complex and non-linear problems, while the usage of a non-linear transfer function in the output layer fails to improve performance targets. Neuron activation depends on its connection weight and

Table 7
Non-dimensional SIFs Y_I variation with y and α for a (10,10,10)_BR network (x = 37.5 mm, $\beta = 0°$). The corresponding analytical value obtained in [29] is shown for comparison in parentheses.

$\alpha[°]y[mm]$	0	5	10	15
0	2.26 (2.26)	2.28 (2.26)	2.29 (2.27)	2.32 (2.3)
10	2.2 (2.2)	2.29 (2.28)	2.38 (2.37)	2.49 (2.48)
20	2.09 (2.08)	2.25 (2.23)	2.41 (2.4)	2.59 (2.58)
30	1.92 (1.91)	2.13 (2.12)	2.36 (2.36)	2.62 (2.61)
40	1.67 (1.67)	1.95 (1.95)	2.24 (2.25)	2.56 (2.56)
50	1.38 (1.39)	1.7 (1.72)	2.05 (2.07)	2.42 (2.43)
60	1.05 (1.04)	1.41 (1.42)	1.8 (1.81)	2.21 (2.22)

Table 8
Non-dimensional SIFs Y_{II} variation with y and α for a (10,10,10)_BR network (x = 37.5 mm, β = 0°). The corresponding analytical value obtained in [29] is shown for comparison in parentheses.

$\alpha[°]y[mm]$	0	5	10	15
0	0.01 (0)	0.3 (0.03)	0.07 (0.06)	0.09 (0.08)
10	0.22 (0.22)	0.25 (0.25)	0.27 (0.29)	0.3 (0.31)
20	0.43 (0.43)	0.45 (0.47)	0.48 (0.5)	0.52 (0.54)
30	0.62 (0.62)	0.64 (0.66)	0.67 (0.7)	0.71 (0.75)
40	0.79 (0.8)	0.81 (0.84)	0.84 (0.88)	0.88 (0.93)
50	0.94 (0.95)	0.96 (0.99)	0.98 (1.04)	1.02 (1.09)
60	1.07 (1.08)	1.08 (1.11)	1.1 (1.16)	1.14 (1.21)

bias value, determined by a specific training algorithm, where the target values are optimized to meet the original outputs (backpropagation). Once fully trained, the ANN is capable of immediate assessments.

In this paper, several ANN, with different architectures, were used to interpolate the mode I and II non-dimensional stress intensity factors (SIF). The original FEM results were organized using four input variables, x and y cartesian coordinates of the specimen crack tip, β the slope at its tip and the specimen loading direction,α. The two outputs Y_I and Y_{II} can be used to calculate the SIFs using equations (5) and (6). Table 1 shows the input variables lower and upper bounds, as well as the data increment used on the training data. The correct input variable definition is very important for ANN training, especially their independence [56]. Therefore, α was chosen as an input variable, instead of $\alpha - \beta$ differential. Both input and output data were normalized before being fed to the ANN model. Each variable was mapped to [-1,1] using a linear function and the necessary data (maximum and minimum values) for denormalization is stored. This pre- and post-processing allows for the network training not to be influenced by input or output absolute values.

The networks were developed using MATLAB's neural net fitting tools. Three different network configurations were analysed: a shallow ML model, with one hidden layer, and two DL models, with two or three hidden layers. DL solutions with more hidden layers were considered but without further results enhancements. Table 2 shows the ANN configurations. A maximum of 60 neurons was used on the shallow network architecture, while a maximum of 45 neurons was used on the three hidden layers DL solution. For two and three hidden layers networks, Table 2 indicates the neurons' distribution per layer and the total number of neurons per network.

The minimum neuron number on each hidden layer was 5, and the total number of neurons was incremented (using a 5 neuron increment scheme) until the network performance converged to a minimum value. The network performance was assessed using the root mean squared error (*RMSE*):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x_t)^2} \tag{10}$$

where x_i is the network prediction, x_t is the original target for the same input variables and n is the total number of data points. Backpropagation algorithms typically divide the available data into training, testing and validation data. Training data is used for backpropagation, validation data for early training stop and testing data for further checking that the network is generalizing well. In this paper, data division was random, using 70 % of data for training, 15 % for validation and 15 % for testing. The 1120 sets of parameters presented in Appendix A of the previous study [29], which is the total number of different combinations from Table 1, were considered as input data. The initial values for the network weights and bias were also considered randomly assigned. This can lead to different network performance and predictions' variability [57]. Several authors deal with this problem by training the same architecture several times and taking the result as the average of all training predictions. Therefore, in this paper, each network was trained five times, stored, and then used to predict each point five times, before taking their average.

Finally, two backpropagation algorithms were used, Levenberg-Marquardt (LM) and Bayesian regularization (BR). Both use the Levenberg-Marquardt algorithm [58], but while the LM algorithm uses early stop training conditions to avoid overfitting, the BR algorithm does not imposes stopping conditions, enhancing the network to accurately predict targets from the training pool. Overfitting may occur if the training process enables the model to learn the training data behaviour and noise so thoroughly, that it will perform poorly on new and unseen data. Overfitting may occur due to model complexity (excessive number of hidden layers or neurons); insufficient training data; high levels of training data noise; long training sessions; or lack of regularization. These may lead to large network weights, adjusting the network to the specifics of the training data. Both LM and BR algorithms use regularization techniques to produce a smoother network response. Regularization is typically performed by adding an additional term to the objective function that penalizes large network weights, while early stopping has been seen as an effective way to reduce the occurrence of large bias [59]. During training, each data point will perform several forward and backward network passes, providing the necessary information for weight and bias optimization. Once every point has performed a forward and backward pass, a full training cycle or epoch has passed. The necessary number of training epochs depends on the training algorithm and the data nature. A trained network is capable of accurate predictions.

4. AI results

4.1. Network performance

As mentioned, network performance was measured using the predicted values vs. original targets RMSE. The parameters of the ANN models studied here are the number of hidden layers, the number of neurons per layer and the backpropagation algorithm. Fig. 3a (blue dots) shows the RMSE evolution vs. the number of neurons used in a shallow learning model, trained with the LM algorithm. Network performance increases with the number of neurons, as RMSE decreases from 0.098 to 0.017. Besides, there is no significant performance gain when using more than 45 neurons. Fig. 3b (orange dots) shows a similar behaviour when training similar networks using the BR algorithm, now with RMSE reaching a minimum of 0.009 when using 60 neurons. In order to train the ANN, 70 % of the original data is randomly chosen as training points. Without early stop conditions, the BR algorithm is capable of accurately capturing the behaviour of the data points, reducing the network RMSE, even if the performance on the validation and testing subsets decreases.

Fig. 4a shows the regression between the networks' output and the target values for a shallow learning network with 5 neurons. Points were distributed between training, testing and validation points according to the LM algorithm. This number of neurons it not enough to capture the data behaviour, resulting in a low correlation factor, and a considerable point dispersion. On the other hand, Fig. 4b shows the regression results when using 45 neurons on the hidden layer. Now, the network is able to

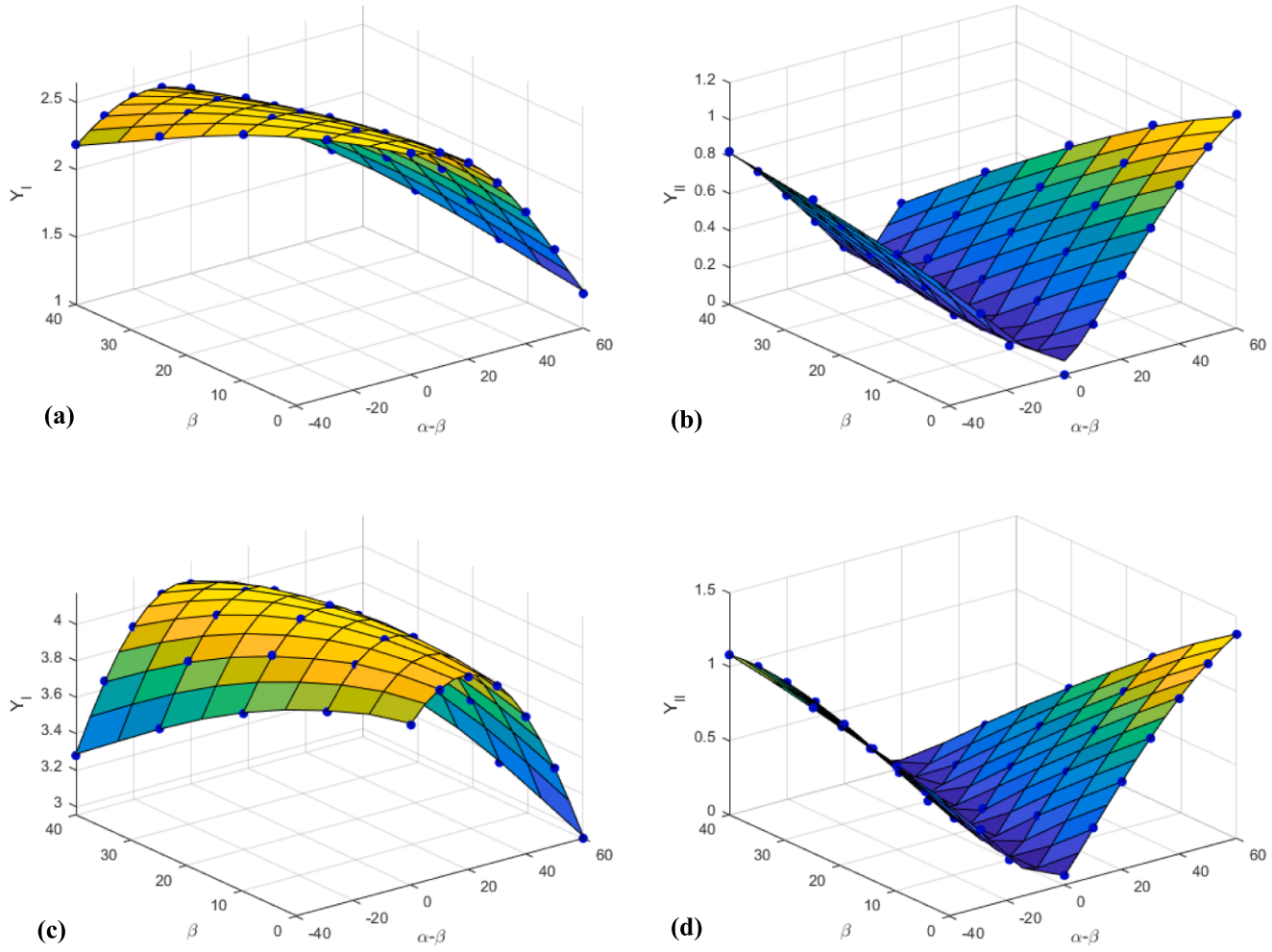


Fig. 11. Non-dimensional SIF (a) Y_I and (b) Y_{II} variation with β and α −β when x = 42.5 mm, y = 0 mm; (c) Y_I and (d) Y_{II} variation with β and α −β when x = 52.5 mm, y = 10 mm for a (10,10,10)_LM network.

capture the behaviour of the original data, increasing the correlation factor and reducing the overall dispersion.

Fig. 5show the difference between the backpropagation algorithms regarding the network performance, using a network with 45 neurons. While Fig. 5a shows that training was stopped around epoch 55, as the networks' performance on the validation subset has reached its minimum, when using the LM algorithm, Fig. 5b shows that the BR algorithm does not stop the training, even if the performance on the testing subset decreases. Training was eventually stopped at epoch 1000 to avoid overfitting problems. It should be mentioned that MATLAB uses the mean squared error (MSE) as its internal loss function.

Increasing the number of hidden layers allows to develop more complex DL models. Fig. 6a (blue dots) and 6b (orange dots) show the RMSE evolution with the ANN total number of neurons for two hidden layers networks, trained with the LM and BR algorithms respectively. Increasing the number of neurons, increases the network performance, decreasing the RMSE to 0.014, when using the LM algorithm, or to 0.006, when using the BR algorithm. These results were obtained with the (25,25) network. When using only 5 neurons on each hidden layer, the best network performance was obtained with the LM algorithm. Using two hidden layers, more complex network architectures can be found. Increasing the number of neurons on the first hidden layer (e.g. (10,5), (15,5) or (15,10)), increases the network performance when compared to their counterparts (e.g. (5,10), (5,15) or (10,15)). Yet, for a similar total number of neurons, the networks with better performance displayed the same number of neurons on both hidden layers (e.g. (10,10), (15,15), (20,20)).

Finally, Fig. 7a (blue dots) and 7b (orange dots) show a similar behaviour for DL networks with three hidden layers. The RMSE decreases to 0.013 for the 15 × 15 × 15 network trained by the LM algorithm or to 0.007 when using the BR algorithm. Fig. 7a (blue dots) shows that when training the networks with the LM algorithm, their performance increases with the number of neurons, but quickly reaches its maximum, and no significant differences were found between the different neuron distribution on the hidden layers. Using the BR algorithm, Fig. 7b (orange dots) shows a clear tendence for the RMSE to decrease with the number of neurons, and once again using a higher number of neurons on the first or second hidden layer, improves networks' performance.

Overall, Fig. 6c shows a clear improvement on the network performance when using two hidden layers vs. only one, especially when using the BR training algorithm. On the other hand, Fig. 7c shows that no improvement was found when using three hidden layers vs. two layers. In fact, while the two hidden layers models using the BR algorithm performed better than the three hidden layers counterparts, no differences were found between the models trained with the LM algorithm.

The trained networks were capable of SIFs prediction for different values of x, y crack tip coordinates, β crack tip angle and α loading angle.

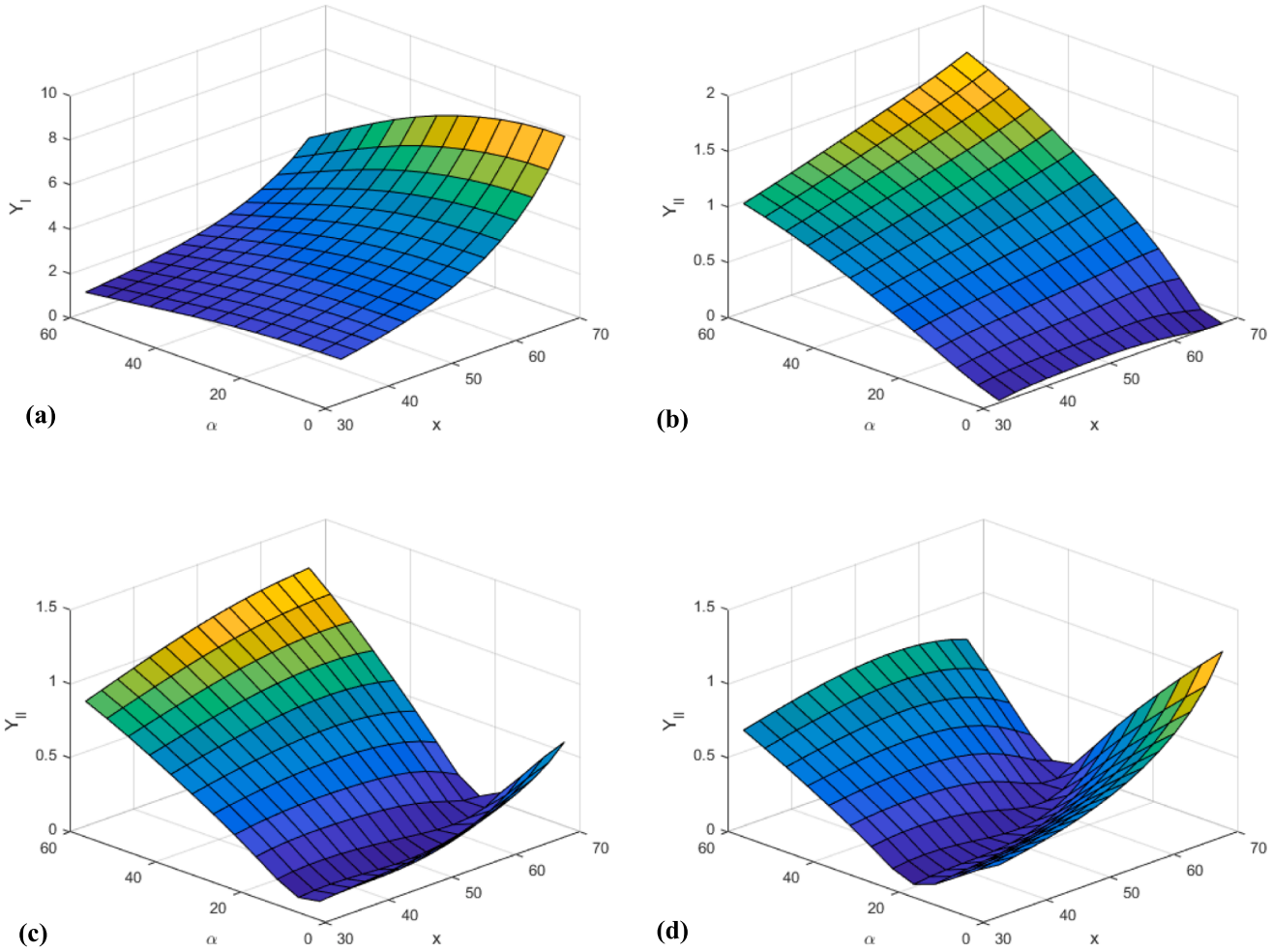


Fig. 12. Non-dimensional SIF (a) Y_I and (b) Y_{II} variation with α and x when y = 0 mm, β = 0°; (c) Y_{II} variation with α and x when y = 5 mm, β = 10°; and (d) Y_{II} variation with α and x when y = 5 mm, β = 20° for a (10,10,10)_LM network.

The lowest RMSE obtained was 0.006, with a (20,20) two-layer network, trained by the BR algorithm or 0.013, with a (10,10,10) three-layer network, trained by the LM algorithm. When compared against the analytical solution, these results are an extraordinary improvement, as RMSE of the analytical solution was 0.052. Due to its faster training time, the (10,10,10)_LM network was predominantly used on the following sections.

Once trained, an ANN can also be provided as a series of parameters, allowing for direct SIFs predictions. Fig. 8 shows the (10,10,10)_LM networks' architecture. The networks' parameters can be found in Annex A, which enables the network reconstruction in MATLAB.

4.2. Non-dimensional stress intensity factors

Once trained, an ANN was able to predict the non-dimensional SIFs for a given x,y crack tip coordinates, β crack tip angle and α loading angle. Fig. 9 compares the performance of a network with one hidden layer with 5 neurons vs. network with three hidden layers and 10 neurons per layer. Both networks were trained with the LM algorithm. Fig. 9a and b show that the 5 neurons network is not able to accurately predict (underfitting) the non-dimensional SIF for this crack location. On the other hand, Fig. 9c and d show that when using a three hidden layer network with a total of 30 neurons (10,10,10), both non-dimensional SIF are accurately predicted. There are also no signs of overfitting when using this ANN. The network was tested using the same region of interest, but the pooling increment was reduced by a factor of 0.5, therefore between each training point a new testing point was calculated, with the non-dimensional SIF behaviour being accurately captured.

Fig. 9 displays the non-dimensional SIFs for a crack located in x = 52.5 mm and y = 0 mm, and their typical variation with α and β. The mode I geometric factor Y_I decreases with α (for a crack aligned with the CTS specimen $\beta = 0^{\circ}$) as the mode mixity [52,60]:

$$M_e = \frac{2}{\pi} tan^{-1}\left|\frac{Y_I}{Y_{II}}\right| \tag{11}$$

decreases and the crack tendency to deflect increases. For inclined cracks ($\beta > 0^{\circ}$), Y_I may increase with α, as the crack may not tend to deflect for mode I dominated loadings. Fig. 9b and d show a clear inflection point for the mode II geometric factor Y_{II}. This point marks the transition between a clockwise or counterclockwise crack deflection. For a crack aligned with the CTS specimen, Y_{II} always increases with α, as the crack will deflect counterclockwise. But as the crack slop increases, crack deflection will occur in the opposite direction, as the crack once again aligns with the specimen. According to Antunes *et al.*, this transition occurs when $\alpha - \beta \approx 0$ [29].

Fig. 10 analyses Y_I and Y_{II} variations with the crack tip vertical location y and the loading angle α. The crack tip is located at x = 37.5

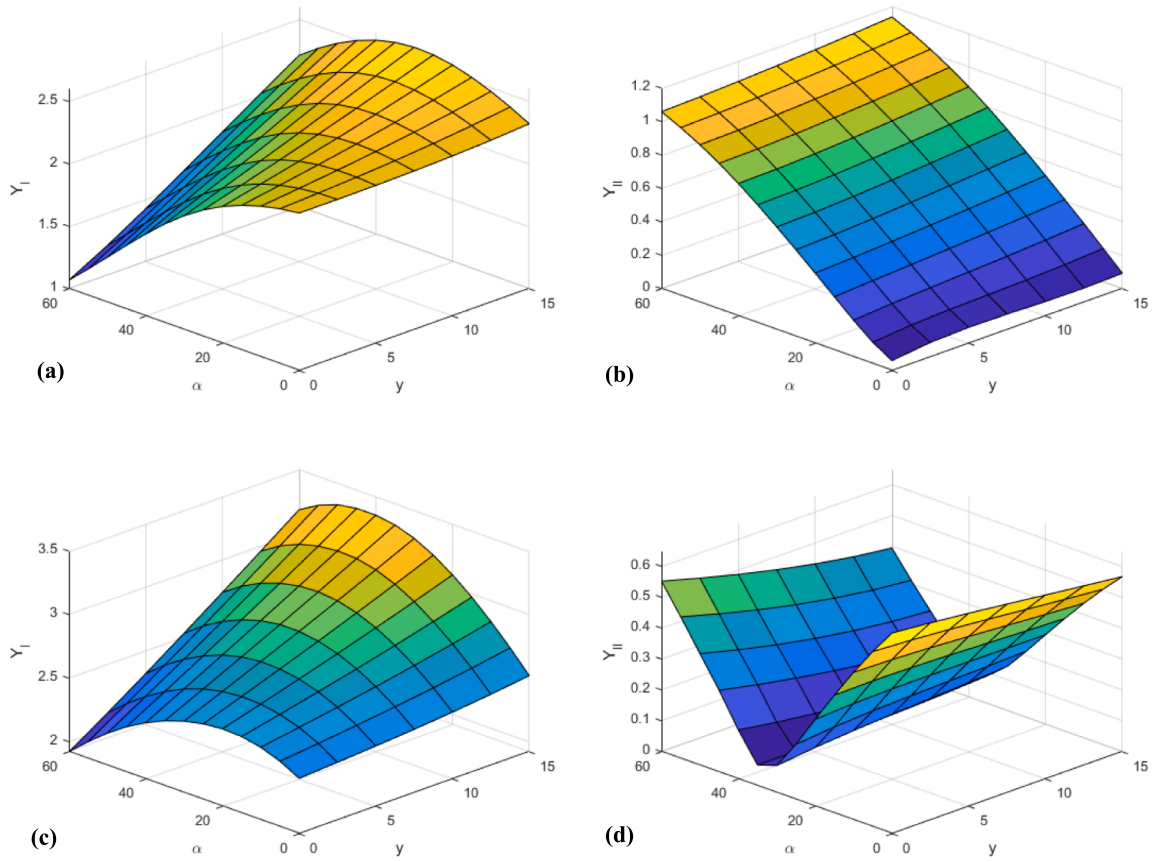


Fig. 13. Non-dimensional SIF (a) Y_I and (b) Y_{II} variation with α and y when x = 37.5 mm, β = 0°; (c) Y_I and (d) Y_{II} variation with α and y when x = 42.5 mm, β = 30° for a (10,10,10)_LM network.

mm and the crack slope is 0°. Fig. 10a and b were obtained for a (10,10,10) network trained with the LM algorithm, while Fig. 10c and d were obtained with the same network trained with the BR algorithm. An example of overfitting can be found in Fig. 10d (black arrow), where the network trained using the BR algorithm over adapts to some variation present in the Y_{II} values when $\alpha = 0^{\circ}$ ($Y_{II} \approx 0$).

Y_I behaviour is barely affected by the crack location, if a pure mode I loading is applied, while for mixed mode loadings Y_I increases with y, reducing the crack deflection effect (Fig. 10a and c). As the mode mixity is decreased, Y_{II} increases and Y_I decreases, increasing the tendency for crack deflection. Fig. 10b and d also show a reduced effect of the crack vertical location over Y_{II}.

In order to analyse the differences between the new model with the previously developed analytical solution [29], Table 3 and Table 4 show the values of Y_I and Y_{II}, respectively, when $x = 52.5$ mm and $y = 0$ mm using a (5)_LM network. Table 3 shows some differences for Y_I between the ANN and analytical (in parentheses) solutions. These tend to increase with both α and β, with the major differences occurring when $\alpha = 60^{\circ}$ and $\beta = 0^{\circ}$ (7 %) or when $\alpha = 50^{\circ}$ and $\beta = 40^{\circ}$ (6 %). A similar behaviour was verified in Fig. 9a, when the ANN model is compared against the original data. For the considered points, the analytical solution performs better than the ANN solution (0.9 vs. 1.9 % average error). The same analysis was performed in Table 4 for Y_{II}, with some differences increasing up to 150 % as $\alpha - \beta \approx 0^{\circ}$ (note that the percentual difference was not calculated if $Y_{II} \approx 0$). Once again, the analytical solution performed better than the ANN model (4 vs. 25 % average error) when compared to the original data.

Increasing the number of hidden layers and neurons per layer, the (10,10,10)_LM network was able to improve the predicted values of Y_I and Y_{II} (Table 5 and Table 6), surpassing the performance of the analytical solution, for the points considered in Fig. 9c and d. Table 5 shows the ANN solution was able to predict the considered Y_I values with an average error of 0.3 % when compared against the original data. When compared against the analytical solution, the (10,10,10)_LM network maintains the same behaviour as the (5)_LM network, with the maximum difference for Y_I dropping to 4 % and for Y_{II} to 13 %.

Finally, Table 7 and Table 8 show the values of Y_I and Y_{II}, respectively, when $x = 37.5$ mm and $\beta = 0^{\circ}$ using a (10,10,10)_BR network. The different backpropagation algorithm allows for a better adaptation of the network to the original data. When this network is compared against the analytical solution, Y_I displays a maximum difference of 1 % when $\alpha = 60^{\circ}$ and $y = 10$ mm, while for Y_{II} this maximum is 6 % when $\alpha = 60^{\circ}$ and $y = 15$ mm. Overall these differences increase both with α and y (disregarding the points where $Y_{II} \approx 0$). When both solutions are compared with the original data, the BR algorithm allows for a better performance than the analytical solution, with the average error decreasing from 0.4 to 0.3 % for Y_I, and from 7.2 to 2.1 % for Y_{II}.

Using a three-layer DL network with 10 neurons on each layer, trained with the LM algorithm (10,10,10), Fig. 11 shows the non-dimensional SIFs variations with β and $\alpha - \beta$. For a crack tip located at $x = 42.5$ mm and $y = 0$ mm, Fig. 11a and b show Y_I and Y_{II} predicted behaviour when compared to the original training points (blue dots). While some differences may be found if $\beta = 0^{\circ}$ and $\alpha - \beta = 0^{\circ}$, the network predictions are in excellent agreement with the original FEM solution. While the Y_I behaviour has been previously described, using $\alpha - \beta$ it is now clear that when this difference is zero, no crack deflection will occur ($Y_{II} = 0$). Considering an initial crack aligned with the specimen, crack deflection will occur counterclockwise, and the crack

tip vertical location will increase. Fig. 11c and d show similar behaviour for a crack tip located at $x = 52.5$ mm and $y = 10$ mm. Clockwise crack propagation with deflection will now occur for $\alpha - \beta \geq 0^{\circ}$, therefore a lower mode mixity is required for counterclockwise crack deflection. A typical crack will then deflect on the opposite direction and align itself with the CTS specimen.

Fig. 12 shows the Y_I and Y_{II} variation with α and the crack tip x coordinate. Fig. 12a and b represent a crack with $y = 0$ mm and $\beta = 0^{\circ}$, or an initial crack aligned with the CTS specimen. Fig. 12a shows that Y_I increases with the initial crack length but decreases with the mode mixity introduced by α. Fig. 12b shows the network has some difficulty in predicting $Y_{II} = 0$ when $\alpha = 0^{\circ}$, but is able to predict Y_{II} increase with α and the crack length. As the crack propagates, Fig. 12c represents a crack tip located in $y = 5$ mm with a 10° slope, while Fig. 12d represents the same crack location but with an increase slope of 20°. While no significant differences were found for Y_I, Y_{II} predicted distribution shows that the crack can now deflect on the opposite direction, for lower values of α, or continue to deflect counterclockwise until $\alpha - \beta \approx 0^{\circ}$.

Fig. 13 shows the influence of α and the crack tip vertical position y, over Y_I and Y_{II}. Fig. 13a and b represent a crack with a zero slope and $x = 37.5$ mm, and their results have been described in Fig. 10c and d. But as the crack length and slope increases Y_I behaviour is now more complex, increasing and decreasing with α (Fig. 13c), while Y_{II} shows that crack deflection can occur in both directions depending on the applied mode mixity. Fig. 13d also shows that for $\beta = 30^{\circ}$ Y_{II} slightly decreases with y. Once again crack deflection behaviour transition occurs when $\alpha - \beta \approx 0^{\circ}$.

These results are in line with Khatammanesh *et al.* [61]. These authors have experimentally and numerically determined the mode I and II SIF evolution for different loading angles on a square CTS specimen. For a loading angle $\alpha = 60^{\circ}$, this specimen renders $Y_I/Y_{II} = 2.5$, while our results show that $Y_I/Y_{II} = 1.5$.

5. Discussion

One important issue is the capability of researchers to use AI, apart from those who developed the solution. The closed-form SIFs' solutions can easily be implemented by different researchers, namely those who develop experimental work. Recently many authors have also used ML tools to derive effective closed-form solutions to complex problems. Florio *et al.* used symbolic regression (SR) to derive solutions for chaotic systems, as the available data and SR are able to provide grey-box solutions with superior interpretability. The developed models, in turn, allow for system behaviour evolution prediction [48]. Complex material behaviours, such as hyperplastic models, can also be analysed using SR, with Hou *et al.* deriving new models and their equations using SR, achieving correlation factors of 0.99 [62]. This was possible by decoupling the high-dimensional problem into simpler sub-problems with the help of AI tools [63]. As mentioned, in the fracture mechanics area, Merrell *et al.* used SR to derive equations for the SIF on an elliptical crack. These authors were able to reduce the Raju-Newman equations complexity by 50 %, while maintaining its performance, or match the original solution complexity, while decreasing the average error from 1.6 to 0.8 %. They also developed a balanced solution with a complexity of 39 %, while providing a solution with an average error of 1.1 % when compared with the original data [49]. This approach may prove to be an effective solution to derive new closed-form equations in the future.

Although many users still consider AI models to be a black box [30], this paper provides important insights on the usage of ANN in the prediction of SIF for mixed-mode fracture mechanics specimens. The use of ANN model to predict non-dimensional SIFs proved to be a useful tool, as the 39 (for mode I) and 54 (for mode II) analytical constants, can be replaced by a given number of parameters defining the ANN architecture. In order to be used as an out-of-the-box solution, Annex A, provides the weights matrices and bias vectors, for a (10,10,10) DL network. This network can be used without the need for the training stage. Otherwise, the network parameters (the neurons weights and bias) can be determined using a training algorithm. Both the LM and BR backpropagation algorithms proved to be effective. The LM algorithm was faster but less effective. The early stop conditions, allow for the training to stop around 60 epochs, while the BR usually runs until the maximum considered number of training epochs (1000). Some signs of overfitting were found in networks trained with the BR algorithm, due to excessive adaptation to the dataset small variations when $Y_{II} \approx 0$. Bahrami *et al.* [32] also used these algorithms to train ANN able to predict specimens' fracture loads under mixed mode conditions. The BR trained models also showed the lowest errors between predictions and their targets, resulting in the most accurate AI models. Due to the fact that network initialization is always random (different initialization weights and bias values, and different points subsets division), ANN prediction is subjected to some variation. The present work shows that while one prediction might not be always accurate, taking the average result of several trainings, does provide accurate results [57]. Present results also show that in order to capture the SIFs' behaviour, an ANN model must possess a minimum number of neurons [33]. As Fig. 4a shows, a larger point dispersion on the model regression fit, corresponds to an underfitting scenario (Fig. 9a and b). Underfitting occurs when the network is not able of capturing the data nature [42]. Our results show that at least 30 neurons are required for all types of analysed networks to accurately predict SIFs' behaviour (preferably evenly distributed between all hidden layers).

In this paper one may consider that a grid-based approach to hyperparameter optimization (HPO) was performed. Uzun *et al.* used a similar approach to analyse the influence of network hyperparameters on ANN models to predict nuclear reactor channel temperatures. While the goal of these authors was not to optimize the network architecture, they concluded that a higher learning rate combined with more than 2 hidden layers and at least 32 neurons per layer provided ANN models with accurate temperature predictions and less training fluctuation [64]. Yet, like on the current paper, the number of hyperparameters analysed was small. When dealing with a larger number of parameters to be optimized, using a structured approach is fundamental. Soares *et al.* used a combination of the particle swarm with Bayesian optimization to dynamically tune the hyperparameter of an ANN bale to classify bearing failure. Overall, the developed approach combined *meta*-heuristic and statistical modelling optimization, increasing the classification accuracy up to 97.5 % and improving up to 11 % the performance of networks with empirically determined hyperparameters [46]. While providing improved results, there are still several challenges to solve regarding HPO methodologies: imbalanced data may lead to erroneous parameter tuning and overfitting; small datasets may not provide the necessary data for effective HPO; inappropriate HPO methodology selection may decrease network accuracy; and time and computational resource costs may affect ANN usage [43].

The main application of the ANN models developed in this paper can be fracture load calculation [32], or crack propagation rate calculation [65,66]. This usually implies that crack propagation must be performed incrementally [11,67,68]. According to the algorithm developed by Baptista *et al.* [34] the SIF are evaluated by FEM for each increment, and the crack increment and propagation direction are calculated. Then the geometry model is remeshed and the crack front updated. The SIF map developed by Antunes *et al.* [29] can be used to estimate the SIFs for a give crack location and slope, without the need to solve a FEM problem. Crack propagation according to different criteria can be assessed easily, and with the work developed in this paper, these results can be used without the need for an analytical solution determination. While crack propagation can also be estimated with AI tools based of full paths [34,36,37], the authors believe this approach can further reduce the number of FEM simulations needed for ANN training.

6. Conclusions

The knowledge of stress intensity factor, K, is important for the study of fracture and fatigue of materials. Usually K is presented in closed-form solutions, however, in situations involving a large number of parameters, accurate solutions may be difficult to obtain in all parametric space. Therefore, AI is emerging as an interesting alternative, which was applied here to study mode I and mode II stress intensity factors in CTS specimens. The main conclusions are:

– Accurate values of geometric factors, Y_I and Y_{II}, were obtained within the region of interest, with an accuracy up to 90 % better when compared with the closed-form solutions proposed in a previous work [29]. This raises the quality of fracture and fatigue analysis made with the CTS specimen.

– From the analysed ANN architectures and training algorithms, the total number of neurons increased the network performance but increased to training times, while the BR backpropagation algorithm provided better performances but longer training times;

– When using the LM training algorithm, network performance improves until about 30 neurons and three hidden layers are used. These networks also showed enhanced abilities to prevent overfitting;

– When using the BR training algorithm, performance converges for a two hidden layers network with 40 neurons, and the use of an extra hidden layer led to overfitting behaviour, with the network over adapting to small data variation;

– The provided network weights and bias allow for network usage without the need for new training tasks;

– The analysis of K solutions using AI is very promising for studying complex situations, namely involving non-homogeneous materials.

CRediT authorship contribution statement

R. Baptista: Writing – review & editing, Validation, Methodology. **V. Infante:** Writing – review & editing. **L.F.P. Borrego:** Writing – review & editing. **E.R. Sérgio:** Writing – review & editing, Data curation. **D.M. Neto:** Writing – review & editing, Formal analysis. **F.V. Antunes:** Writing – original draft, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors gratefully acknowledge the financial support from the Portuguese Foundation of Science and Technology (FCT) under the projects UIDB/00285/2020 and LA/P/0112/2020. Edmundo Sérgio is also grateful to the FCT for the PhD grant with reference 2022.11438.BD. This research was also funded by the FACI-2023-MPR-04-014323 project, supported by European Funds and facilitated by Agência Nacional Inovação (ANI) under the Eureka smart label S0612-BRACER. This research was also funded by project MATEL, with the reference COMPETE2030-FEDER-00587200, supported by European Funds under Portugal 2030, Compete2030 and FEDER.

Appendix A

Table A1
First hidden layer neurons' weights matrix for (10,10,10)_LM network.

Input 1	Input 2	Input 3	Input 4	Input / Output
−0.6969	−0.0066	0.0690	0.0559	Neuron 1
−0.7394	−0.0793	0.3223	−0.5058	Neuron 2
−0.1726	−0.1743	0.3810	0.1197	Neuron 3
−0.6994	−0.1448	0.8673	−0.9381	Neuron 4
0.0029	−0.2542	−0.1450	−0.0369	Neuron 5
0.0200	−0.0034	0.2876	−0.1574	Neuron 6
−0.8432	0.0058	4.1384	−4.0538	Neuron 7
−0.2650	0.1783	−0.4879	0.0933	Neuron 8
−0.0766	−0.3127	−0.2587	−0.5983	Neuron 9
−1.1114	−0.3180	0.4840	−0.0290	Neuron 10

Table A2
Second hidden layer neurons' weights matrix for (10,10,10)_LM network.

Neuron 1	Neuron 2	Neuron 3	Neuron 4	Neuron 5	Neuron 6	Neuron 7	Neuron 8	Neuron 9	Neuron 10	Input / Output
0.1876	0.8718	0.1946	−0.3663	−1.2570	−0.7818	−0.0466	−0.0951	0.6017	−0.0992	Neuron 1
−0.6435	−1.0733	0.5018	0.8684	0.1101	−0.0899	0.3195	−0.7602	0.2386	0.0882	Neuron 2
−0.6434	1.7884	0.7848	0.1422	−0.8239	0.6333	0.4772	−0.7973	0.9458	−0.5012	Neuron 3
−1.3094	0.2332	−0.3255	0.3852	−0.0345	2.1655	0.2031	−0.5122	0.4167	0.2026	Neuron 4
0.3265	−0.0932	−1.0522	0.0417	−0.3965	−1.5088	0.0715	0.4377	−0.0637	0.2281	Neuron 5
−1.2019	1.0125	1.6096	0.7110	−0.9662	−0.2872	0.5166	−0.1600	0.8358	−0.2302	Neuron 6
1.5709	0.0083	0.5002	−0.0466	0.3651	1.1084	−0.0258	0.5440	0.0230	−0.0701	Neuron 7
−1.8653	0.1614	−0.3927	0.5747	0.2136	0.5313	0.2853	−0.2153	0.0258	0.1351	Neuron 8
−0.4048	0.1358	0.3030	−0.3338	−0.3365	0.0357	−0.3893	−1.0237	−0.8245	0.7630	Neuron 9
−1.6214	1.7399	−1.4107	−0.2295	−0.6146	−2.0804	0.2067	−0.8725	−0.0121	0.3888	Neuron 10

Table A3
Third hidden layer neurons' weights matrix for (10,10,10)_LM network.

Neuron 1	Neuron 2	Neuron 3	Neuron 4	Neuron 5	Neuron 6	Neuron 7	Neuron 8	Neuron 9	Neuron 10	Input / Output
−0.0738	−0.9414	−0.8109	0.0274	−0.7491	0.9194	−1.5969	0.6371	−0.1395	1.2641	Neuron 1
−0.1939	−1.1776	−0.8877	−0.2035	−0.5862	0.9913	1.3563	0.7068	−0.7392	0.3366	Neuron 2
0.6076	−0.9665	0.4911	−0.2391	−1.4151	−0.6546	−0.3988	1.4692	−0.4814	1.3180	Neuron 3
−0.4513	1.0326	1.3785	0.1701	0.4325	−0.3940	−0.0426	−1.5672	−0.5692	0.5318	Neuron 4
−0.1983	−0.3681	−0.3119	−0.1064	0.7154	0.2378	2.2196	0.3770	−0.2671	0.1783	Neuron 5
0.2798	−0.4542	0.8352	0.9101	−1.1752	0.4298	0.4383	−0.8990	−1.4246	2.6028	Neuron 6
−0.1185	−0.3853	−0.2550	−0.0686	0.0703	0.2946	−0.8365	0.0835	−0.4955	−0.3580	Neuron 7
−0.1419	1.0646	−0.1414	−1.3562	−1.2891	0.1873	−0.8663	1.5966	−0.6101	1.0110	Neuron 8
1.5025	−1.2640	−0.5212	−0.3955	−0.3364	0.2565	0.0320	−0.1010	0.2785	0.2133	Neuron 9
0.5385	−0.5843	−0.6196	−0.2106	−0.5649	0.0854	−0.4816	1.1709	0.8815	0.8837	Neuron 10

Table A4
Output layer neurons' weights matrix for (10,10,10)_LM network.

Neuron 1	Neuron 2	Neuron 3	Neuron 4	Neuron 5	Neuron 6	Neuron 7	Neuron 8	Neuron 9	Neuron 10	Input / Output
0.4485	−0.5084	0.0656	0.1466	−2.0716	−0.0702	1.0955	0.0004	−0.0065	0.1928	Neuron 1
1.4368	1.5199	2.8301	2.3082	−0.5937	−1.3174	0.9272	−1.5252	−0.7076	−1.4525	Neuron 2

Table A5
(10,10,10)_LM network neurons' bias vectors.

Neuron 1	Neuron 2	Neuron 3	Neuron 4	Neuron 5	Neuron 6	Neuron 7	Neuron 8	Neuron 9	Neuron 10	
1.7522	1.2611	−0.6101	0.7246	0.5139	−0.7484	0.7999	−0.9026	1.4221	−1.2929	Layer 1
−0.6667	0.9236	−1.6544	1.5497	−0.8592	−0.0712	0.2758	1.4028	−1.5631	−2.5132	Layer 2
2.2534	−1.2485	−0.3981	0.5335	−0.2235	0.9235	−0.0947	−0.5423	0.6396	1.3915	Layer 3
1.0092	0.4418									Output

Data availability

Data will be made available on request.

References

[1] N.E. Dowling, Mechanical Behavior of materials: engineering methods for deformation, fracture, and fatigue, Prentice Hall, 2nd Edition,, 1998.
[2] W. Elber, Fatigue crack closure under cyclic tension, Eng. Fract. Mech. 2 (1970) 37–45, https://doi.org/10.1016/0013-7944(70)90028-7.
[3] A. Vasudevan, Application of unified fatigue damage approach to compression–tension region, Int. J. Fatigue 21 (1999) 263–273, https://doi.org/10.1016/S0142-1123(99)00097-3.
[4] A.H. Noroozi, G. Glinka, S. Lambert, A two parameter driving force for fatigue crack growth analysis, Int. J. Fatigue 27 (2005) 1277–1296, https://doi.org/10.1016/j.ijfatigue.2005.07.002.
[5] D. Kujawski, A fatigue crack driving force parameter with load ratio effects, Int. J. Fatigue 23 (2001) S239–S246, https://doi.org/10.1016/S0142-1123(01)00158-X.
[6] A.R. Ingraffea, C. Manu, Stress-intensity factor computations in three dimensions with quarter-point elements, Int. J. Numerical Meth. Eng. 15 (1980) 1427–1445, https://doi.org/10.1002/nme.1620151002.
[7] K.N. Shivakumar, P.W. Tan, J.C. Newman Jr., A virtual crack-closure technique for calculating stress intensity factors for cracked three dimensional bodies, Int. J. Fracture 36 (1988) R43–R50, https://doi.org/10.1007/BF00035103.
[8] F.V. Antunes, J.A.M. Ferreira, J. Byrne, Stress Intensity Factor Calculation Based on the Work of External Forces, Int. J. Fracture 98 (1999) 1–14, https://doi.org/10.1023/A:1018684932071.
[9] J. Triclot, T. Corre, A. Gravouil, V. Lazarus, Key role of boundary conditions for the 2D modeling of crack propagation in linear elastic Compact Tension tests, Eng. Fract. Mech. 277 (2023), https://doi.org/10.1016/j.engfracmech.2022.109012.
[10] R.J. Sanford, J.W. Dally, A general method for determining the mixed-mode stress intensity factors from isochromatics fringe patterns, Eng. Fract. Mech. 11 (1979) 621–633, https://doi.org/10.1016/0013-7944(79)90123-1.
[11] M.R. Ayatollahi, M. Nejati, An over-deterministic method for calculation of coefficients of crack tip asymptotic field from finite element analysis, Fatigue Fract. Eng. Mater. Struct. 34 (2010) 159–176, https://doi.org/10.1111/j.1460-2695.2010.01504.x.
[12] M.L. Williams, On the stress distribution at the base of a stationary crack, Trans. ASME, J. Appl. Mech. (1956) 1–6.
[13] M.S. Reddy, K. Ramesh, A. Thiyagarajan, Evaluation of mode-I SIF, T-stress and J-integral using displacement data from digital image correlation – Revisited, Theor. Appl. Fract. Mech. 96 (2018) 146–159, https://doi.org/10.1016/j.tafmec.2018.04.006.
[14] ASTM E647-15: Standard test method for measurement of fatigue crack growth rates. Philadelphia: American Society for Testing and Materials (2015) ASTM.
[15] H. Tada, P.C. Paris, G.R. Irwin, American Society of Mechanical Engineers, ASM International. The stress analysis of cracks handbook. 3rd ed. New York: ASME Press: Professional Engineering Pub. : ASM International (2000).
[16] Y. Murakami, Stress intensity factors handbook, Oxford [Oxfordshire], Pergamon Press, New York, 1987.
[17] A. Carpinteri, Handbook of Fatigue Crack Propagation in Metallic Structures, Elsevier Science, 1994.
[18] J.M. Alegre, I.I. Cuesta, A. Díaz, Stress-intensity factor solutions for embedded elliptical cracks in round bars subjected to tensile load, Theor. Appl. Fract. Mech. 117 (2022) 103189, https://doi.org/10.1016/j.tafmec.2021.103189.
[19] F.V. Antunes, J.A.M. Ferreira, C.M. Branco, J. Byrne, Stress Intensity Factors for tunneling Corner Cracks Under Mode I Loading, Fatigue Fract. Eng. Mater. Struct. 23 (2000) 81–90, https://doi.org/10.1046/j.1460-2695.2000.00215.x.
[20] J.C. Newman Jr., I. Raju, An empirical stress-intensity factor equation for the surface crack, Eng. Fract. Mech. 15 (1981) 185–192.
[21] J. Newman, I. Raju, Stress-intensity factor equations for cracks in three-dimensional finite bodies, in: Fracture Mechanics: Fourteenth SymposiumVolume I: Theory and Analysis, ASTM International (1983).
[22] P. Jin, Z. Liu, H. Chen, M. Liu, X. Wang, X. Chen, Mixed-mode I&II fatigue crack growth behaviors of 16MND5 steel: The role of crack driving forces and crack closure, Int. J. Fatigue 183 (2024) 108228, https://doi.org/10.1016/j.ijfatigue.2024.108228.
[23] X. Zhang, H. Wang, Q. Lu, S. Hu, Y. Zheng, Fatigue growth behavior of mode II crack in headed stud steel used in steel–concrete composite structures, Eng. Fail. Anal. 161 (2024) 108287, https://doi.org/10.1016/j.engfailanal.2024.108287.
[24] L.W. Wei, L. Edwards, M.E. Fitzpatrick, FE analysis of stress and stress intensity factors of interfacial cracks in a CTS specimen, Eng. Fract. Mech. 69 (2002) 85–90, https://doi.org/10.1016/S0013-7944(01)00072-8.
[25] J. Jamali, A.-H.-I. Mourad, Y. Fan, J.T. Wood, Through-thickness fracture behavior of unidirectional glass fibers/epoxy composites under various in-plane loading using the CTS test, Eng. Fract. Mech. 156 (2016) 83–95, https://doi.org/10.1016/j.engfracmech.2016.01.016.
[26] P. Jin, W. Li, H. Chen, J. Wang, Z. Liu, X. Wang, X. Chen, Compact-tension-shear specimen for orthotropic materials in fracture toughness testing, Theor. Appl. Fract. Mech. 132 (2024) 104488, https://doi.org/10.1016/j.tafmec.2024.104488.
[27] A. Pirondi, G. Nicoletto, Mixed Mode I/II fracture toughness of bonded joints, Int. J. Adhesion & Adhesives 22 (2002) 109–117, https://doi.org/10.1016/S0143-7496(01)00042-2.

[28] M. Sander, H.A. Richard, Experimental and numerical investigations on the influence of the loading direction on the fatigue crack growth, Int. J. Fatigue 28 (5/6) (2006) 583–591, https://doi.org/10.1016/j.ijfatigue.2005.05.012.
[29] F.V. Antunes, R. Branco, J.A.M. Ferreira, L.P. Borrego, Stress Intensity Factor Solutions for CTS Mixed Mode Specimen, Frattura Ed Integrità Strutturale 48 (2019) 676–692, https://doi.org/10.3221/IGF-ESIS.48.64.
[30] H. Wang, B. Li, J. Gong, F.-Z. Xuan, Machine learning-based fatigue life prediction of metal materials: Perspectives of physics-informed and data-driven hybrid methods, Eng. Fract. Mech. 284 (2023) 109242, https://doi.org/10.1016/j.engfracmech.2023.109242.
[31] S. Nasiri, M.R. Khosravani, K. Weinberg, Fracture mechanics and mechanical fault detection by artificial intelligence methods: A review, Eng. Fail. Anal. 81 (2017) 270–293, https://doi.org/10.1016/j.engfailanal.2017.07.011.
[32] B. Bahrami, H. Talebi, M.R. Ayatollahi, M.R. Khosravani, Artificial neural network in prediction of mixed-mode I/II fracture load, Int. J. Mech. Sci. 248 (2023), https://doi.org/10.1016/j.ijmecsci.2023.108214.
[33] V. Giannella, F. Bardozzo, A. Postiglione, R. Tagliaferri, R. Sepe, E. Armentani, Neural networks for fatigue crack propagation predictions in real-time under uncertainty, Comput. Struct. 288 (2023), https://doi.org/10.1016/j.compstruc.2023.107157.
[34] R. Baptista, P. Moita, V. Infante, Fatigue crack growth on modified CT specimens using artificial neural networks, Int. J. Fatigue 167 (2023) 107357, https://doi.org/10.1016/j.ijfatigue.2022.107357.
[35] Z. Huang, D. Chang, X. Yang, H. Zuo, A deep learning-based approach for crack damage detection using strain field, Eng. Fract. Mech. 293 (2023), https://doi.org/10.1016/j.engfracmech.2023.109703.
[36] B. Wang, L. Xie, J. Song, B. Zhao, C. Li, Z. Zhao, Curved fatigue crack growth prediction under variable amplitude loading by artificial neural network, Int. J. Fatigue 142 (2021), https://doi.org/10.1016/j.ijfatigue.2020.105886.
[37] B. Santos, V. Infante, T. Barros, R. Baptista, Study of fatigue crack propagation on modified CT specimens under variable amplitude loadings using machine learning, Int. J. Fatigue 184 (2024), https://doi.org/10.1016/j.ijfatigue.2024.108332.
[38] D. Gope, P.C. Gope, A. Thakur, A. Yadav, Application of artificial neural network for predicting crack growth direction in multiple cracks geometry, Appl. Soft. Comput. 30 (2015) 514–528, https://doi.org/10.1016/j.asoc.2015.02.003.
[39] J.C. Figueira Pujol, J.M. Andrade Pinto, A neural network approach to fatigue life prediction, Int. J. Fatigue 33 (2011) 313–322, https://doi.org/10.1016/j.ijfatigue.2010.09.003.
[40] J.F. Barbosa, J.A.F.O. Correia, R.C.S.F. Júnior, A.M.P. de Jesus, Fatigue life prediction of metallic materials considering mean stress effects by means of an artificial neural network, Int. J. Fatigue 135 (2020) 105527, https://doi.org/10.1016/j.ijfatigue.2020.105527.
[41] Y. Zhao, Y. Liu, Z. Xu, Statistical learning prediction of fatigue crack growth via path slicing and re-weighting, Theor. Appl. Mech. Letters 13 (2023) 100477, https://doi.org/10.1016/j.taml.2023.100477.
[42] W. Liang, M. Lou, Y. Wang, C. Zhang, S. Chen, C. Cui, A fatigue crack growth prediction method on small datasets based on optimized deep neural network and Delaunay data augmentation, Theor. Appl. Fract. Mech. 129 (2024) 104218, https://doi.org/10.1016/j.tafmec.2023.104218.
[43] M.A.K. Raiaan, S. Sakib, N.M. Fahad, A. Al Mamun, M.A. Rahman, S. Shatabda, M. S.H. Mukta, A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks, Decision Analytics J. 11 (2024) 100470, Doi: 10.1016/j.dajour.2024.100470.
[44] D. Iliadis, M. Wever, B. De Baets, W. Waegeman, Hyperparameter optimization of two-branch neural networks in multi-target prediction, Appl. Soft. Comput. 165 (2024) 111957, https://doi.org/10.1016/j.asoc.2024.111957.
[45] G. Békési, L. Barancsuk, B. Hartmann, Deep neural network based distribution system state estimation using hyperparameter optimization, Results in Eng. 24 (2024) 102908, https://doi.org/10.1016/j.rineng.2024.102908.
[46] R.C. Soares, J.C. Silva, J.A. de Lucena, A.C.L. Filho, J.G.G. de Souza Ramos, A.V. Brito, Integration of Bayesian optimization into hyperparameter tuning of the particle swarm optimization algorithm to enhance neural networks in bearing failure classification, Measurement 242 Part A (2025) 115829, Doi: 10.1016/j.measurement.2024.115829.
[47] D. Melching, F. Paysan, T. Strohmann, E. Breitbarth, An iterative crack tip correction algorithm discovered by physical deep symbolic regression, Int. J. Fatigue 187 (2024) 108432, https://doi.org/10.5281/zenodo.10730749.
[48] M. De Florio, I.G. Kevrekidis, G.E. Karniadakis, AI-Lorenz: A physics-data-driven framework for black-box and gray-box identification of chaotic systems with symbolic regression, Chaos Solitons Fractals 188 (2024) 115538, https://doi.org/10.1016/j.chaos.2024.115538.
[49] J. Merrell, J. Emery, R.M. Kirby, J. Hochhalter, Stress intensity factor models using mechanics-guided decomposition and symbolic regression, Eng. Fract. Mech. 310 (2024) 110432, https://doi.org/10.1016/j.engfracmech.2024.110432.
[50] H.A. Richard, A new compact shear specimen, Int. J. Fracture 17 (1981) R105–R107.
[51] J. Setién, J.M. Varona, On the Use of Dimensional Analysis in Fracture Mechanics, in: F. Poitiers, J. Petit, I. Vol (Eds.), Proc. 11th Biennial European Conference on Fracture (ECF11), 1996, pp. 125–132.
[52] P. Jin, X. Wang, H. Chen, Z. Liu, X. Chen, Analysis of mixed-mode Compact-Tension-Shear (CTS) specimens with slanted propagating cracks, Theor. Appl. Fract. Mech. 127 (2023) 104037, https://doi.org/10.1016/j.tafmec.2023.104037.
[53] V. Shlyannikov, E. Martínez-Pañeda, A. Tumanov, R. Khamidullin, Mode I and mode II stress intensity factors and dislocation density behaviour in strain gradient plasticity, Theor. Appl. Fract. Mech. 116 (2021) 103128, https://doi.org/10.1016/j.tafmec.2021.103128.
[54] V.N. Shlyannikov, A.P. Zakharov, Generalization of mixed mode crack behaviour by the plastic stress intensity factor, Theor. Appl. Fract. Mech. 91 (2017) 52–65, https://doi.org/10.1016/j.tafmec.2017.03.014.
[55] H. Yonaba, F. Anctil, V. Fortin, Comparing Sigmoid Transfer Functions for Neural Network Multistep Ahead Streamflow Forecasting, J. Hydrologic Eng. 15 (4) (2010) 275–283, https://doi.org/10.1061/(ASCE)HE.1943-5584.0000188.
[56] S.N.S. Mortazavi, A. Ince, An artificial neural network modeling approach for short and long fatigue crack propagation, Comput. Mater. Sci. 185 (2020) 109962, https://doi.org/10.1016/j.commatsci.2020.109962.
[57] J. Chen, Y. Liu, Fatigue modeling using neural networks: A comprehensive review, Fatigue Fract. Eng. Mater. Struct. 45 (2022) 945–979, https://doi.org/10.1111/ffe.13640.
[58] D.J.C. MacKay, Bayesian Interpolation, Neural Comput. 4 (1992) 415–447, https://doi.org/10.1162/neco.1992.4.3.415.
[59] M. Kayri, Predictive abilities of Bayesian regularization and levenberg-marquardt algorithms in artificial neural networks: A comparative empirical study on social data, Mathematical and Comp. Applications 21 (2016) 20, https://doi.org/10.3390/mca21020020.
[60] M.R. Ayatollahi, S.M.J. Razavi, M.Y. Yahya, Mixed mode fatigue crack initiation and growth in a CT specimen repaired by stop hole technique, Eng. Fract. Mech. 145 (2015) 115–127, https://doi.org/10.1016/j.engfracmech.2015.03.027.
[61] A. Khatammanesh, K. Farhangdoost, D.G. Moghadam, R.M. Nejad, An evaluation of the loading condition on mixed-mode stress intensity factors for CTST specimens made of 2024–T351 aluminum alloy, Eng. Fail. Anal. 130 (2021) 105780, https://doi.org/10.1016/j.engfailanal.2021.105780.
[62] J. Hou, X. Chen, T. Wu, E. Kuhl, X. Wang, Automated data-driven discovery of material models based on symbolic regression: A case study on the human brain cortex, Acta Biomater. 188 (2024) 276–296, https://doi.org/10.1016/j.actbio.2024.09.005.
[63] R. Yi, D. Georgiou, X. Liu, C.E. Athanasiou, Mechanics-informed, model-free symbolic regression framework for solving fracture problems, J. Mech. Phys. Solids 105916 (2024), https://doi.org/10.1016/j.jmps.2024.105916.
[64] S. Uzun, E. Yildiz, H. Arslantaş, Optimizing neural network models for predicting nuclear reactor channel temperature: A study on hyperparameter tuning and performance analysis, Nuclear Eng. and Design 429 (2024) 113636, https://doi.org/10.1016/j.nucengdes.2024.113636.
[65] J.S. Kim, D.J. Kim, S.P. Hong, Effect of Mode II in the mixed-mode on the fatigue crack growth behaviour for SAPH440 material, Int. J. Fatigue 187 (2024) 108470, https://doi.org/10.1016/j.ijfatigue.2024.108470.
[66] G. Lesiuk, M. Smolnicki, R. Mech, A. Zięty, C. Fragassa, Analysis of fatigue crack growth under mixed mode (I + II) loading conditions in rail steel using CTS specimen, Eng. Fail. Anal. 109 (2020) 104354, https://doi.org/10.1016/j.engfailanal.2019.104354.
[67] R. Baptista, V. Infante, J.F.A.A. Madeira, Optimization of a cruciform specimen for fatigue crack growth under in and out-of-phase in-plane biaxial loading conditions, Mech. Advanced Mat. Struct. 30 (8) (2022) 1649–1666, https://doi.org/10.1080/15376494.2022.2038740.
[68] J. Candeias, R. Baptista, R. Cláudio, L. Reis, M. Freitas, On the influence of different in-plane biaxial loading conditions over FCG lives, Int. J. Fatigue 157 (2022) 106714, https://doi.org/10.1016/j.ijfatigue.2021.106714.