

# Determining the defect locations and sizes in elastic plates by using the artificial neural network and boundary element method

Xinyue Han, Yang Yang<sup>\*</sup>, Yijun Liu<sup>\*</sup>

Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen, PR China

## ARTICLE INFO

### Keywords:

Regression inverse problem  
Artificial neural network  
Boundary element method  
Defect localization  
Structural health monitoring

## ABSTRACT

In this work, a new approach for solving the inverse problem of the defect location in an elastic plate is developed by combining the artificial neural network (ANN) and the boundary element method (BEM). The inverse problem is formulated as a regression problem, which extracts the solution of highest probability through machine learning (ML) from a large amount of data. The efficiency and accuracy of data generation is guaranteed by using the BEM, which solves for strain values on the boundaries of plates with circular defects and creates the ML datasets. The ANNs, which are constructed by a fully connected multi-layer perceptron, are trained using these datasets to predict the center coordinates and radii of the circular defects and can achieve about 98% accuracy in the detection. Compared with the Lamb wave based structural health monitoring (SHM) techniques, which in general require signal generation, collection and processing, the proposed approach only requires the data of boundary strains of the considered structures as the input data and a simple training process. Therefore, it is much easier to implement and has great potential in applications of SHM.

## 1. Introduction

Determining the location and size of a defect in a structure is an inverse problem, given the input and output data at some sampling points, such as the stress or strain values, or wave propagation patterns. In contrast with a forward problem, the inverse problem is ill-posed and the solutions are in general unstable and nonunique [1]. It is more difficult to calculate the relevant physical quantities using existing computational methods. Data-driven models based on the machine learning (ML) approach [2–4] are potentially more efficient and accurate than forward-looking predictive models due to the learned probability distribution from analyzing huge volumes of historical data [5–7]. The main advantage of ML is that it exploits the fast data analysis capability of a computer to predict the trend of the physical events [8–10] and to uncover the hidden insights among the enormous amount of seemingly illegible data [11–12]. Thus, it has attracted strong interest in dealing with the inverse problems.

Tamaddon-Jahromi [13] utilized the deep learning methods to test for the linear/non-linear heat conduction, convection-conduction, and natural convection problems, in which the boundary conditions were determined by using the temperature measurements at three, four and five locations. Physics-informed neural networks, a deep learning

framework for solving nonlinear partial differential equations was presented by Raissi et al. [14]. It was later used to address the inverse mechanics problems of thin rectangular plates [15], in which the boundary conditions are to be identified based on the known information including deflection values on part of the domain, basic configuration of the plate and load distribution. Inverse problems of heat conduction with constant and variable parameters were studied by He et al. [16] to predict the unknown thermal conductivity parameters by using coupled physics-informed neural network frameworks with skip connections. Dwivedi et al. [17] proposed a distributed physics informed neural network (DPINN) for solving inverse problems, where they view the unknown model parameters of the PDE as weights of the neural network. The weights were then obtained by minimizing the loss function which contains information from both training data and PDE. Dworakowski et al. [18] adopted artificial neural network ensembles to detect a fatigue damage in aircraft via employing various damage indices. De Fenza et al. [19] utilized a multi-layer perceptron to detect through-thickness holes in metallic and composite plates. Their model was trained with simulation data and tested on experimental data. A novel neural network-based damage localization method was proposed by Zhang et al. [20] and developed for plate-like structures. With the help of the time varying damage index (TVDI) feature extracted from the

<sup>\*</sup> Corresponding authors.

E-mail addresses: [yangy33@sustech.edu.cn](mailto:yangy33@sustech.edu.cn) (Y. Yang), [liuyj3@sustech.edu.cn](mailto:liuyj3@sustech.edu.cn) (Y. Liu).

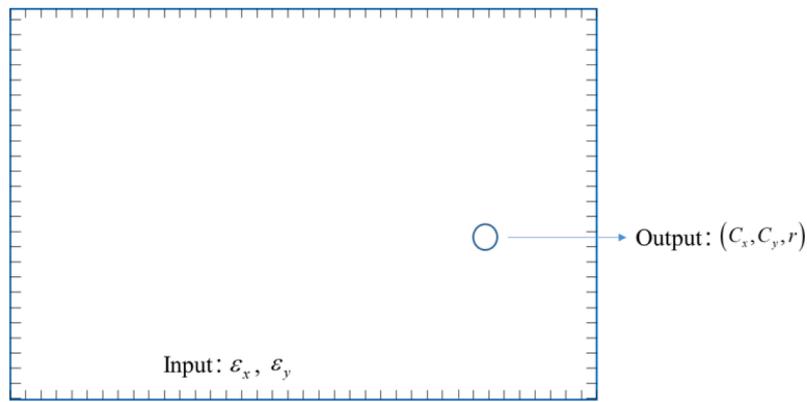


Fig. 1. Input and output of the inverse problem.

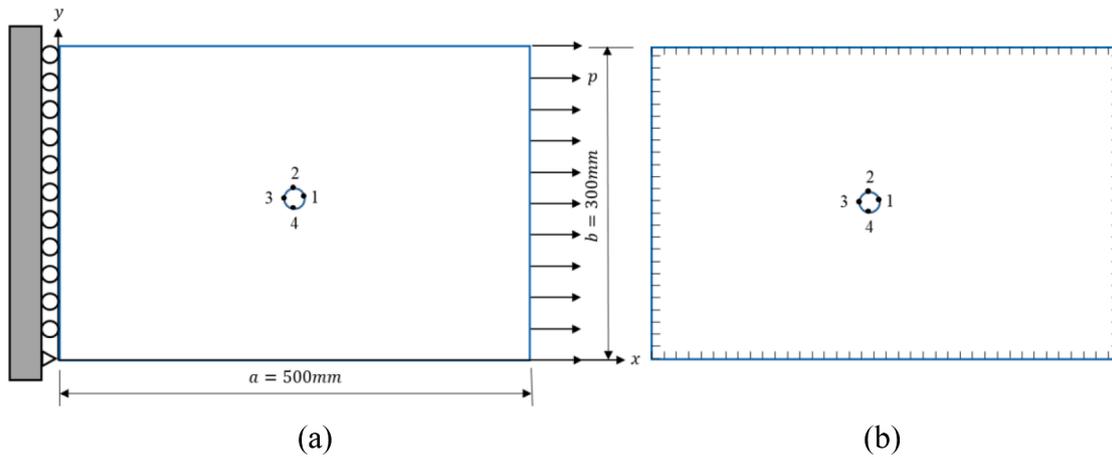


Fig. 2. (a) Geometry and (b) BEM model of the 2D plate with a circular defect.

**Table 1**  
Stresses of the 2D plate with a circular defect as compared with the analytical results.

Node	$\sigma_x$	$\sigma_y$	$\sigma$ (BEM)	$\sigma$ (Exact)	Error (%)
1	0.0000	-0.9836	-0.9836	-1.0	1.64
2	3.0079	0.0000	3.0079	3.0	0.26
3	0.0001	-1.0111	-1.0111	-1.0	1.11
4	3.0064	0.0000	3.0064	3.0	0.21

**Table 2**  
Strains of the 2D plate without a circular defect as compared with the analytical results.

Node	$\epsilon_x$	$\epsilon_y$	$\epsilon_y/\epsilon_x$ (BEM)	$\epsilon_y/\epsilon_x$ (Exact)	$\nu$	Error (%)
1	0.9469	-0.2841	0.3000	0.3	0.3	0.00
2	0.9471	-0.2827	0.2985	0.3	0.3	0.49
3	0.9475	-0.2842	0.3000	0.3	0.3	0.00
4	0.9471	-0.2841	0.3000	0.3	0.3	0.00

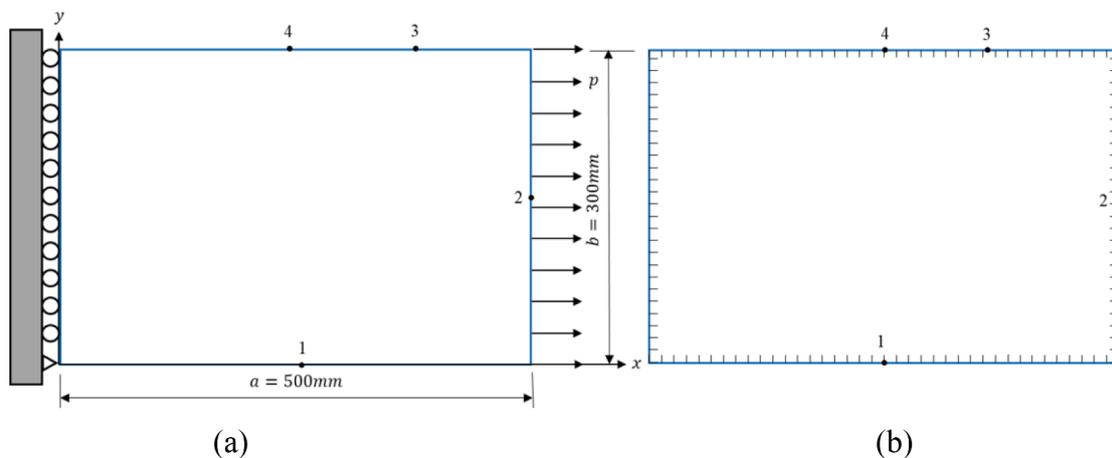


Fig. 3. (a) Geometry and (b) BEM model of the 2D plate without a circular defect.

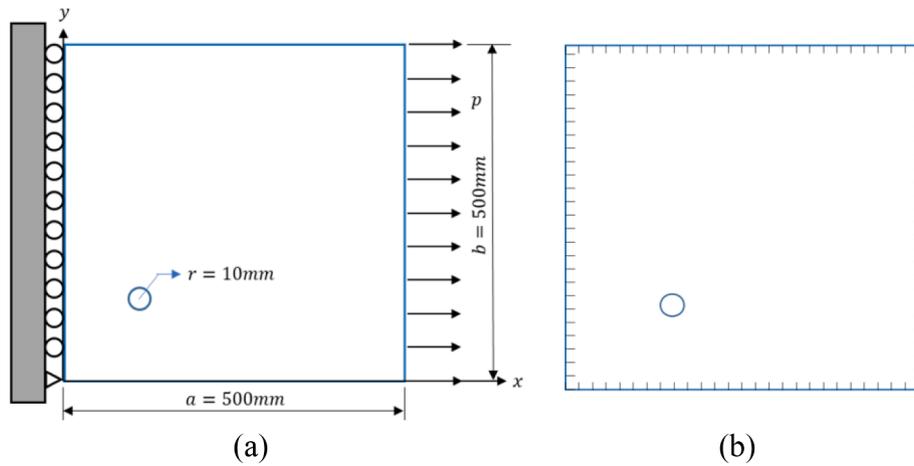


Fig. 4. (a) Geometry and (b) BEM model of the square plate with a circular defect.

Table 3

ANN parameters for case 1.

Number of examples in Training dataset	4592
Number of examples in Testing dataset	1969
Batch-size	32
Number of neurons in the input layer	800
Number of hidden layers	1–3
Number of neurons in the hidden layer	20–200
Number of neurons in the output layer	2
Activation layers	ReLU
Number of epochs	500
Validation split	0.2
Optimizer	RMSprop

Table 4

Model performance with different architectures for case 1.

Neurons at each layer	Trainable parameters	Number of Epochs	Loss value (m)	Accuracy	CPU time (s)
800–100–2	80,302	500	0.0190	0.9512	104
800–200–2	160,602	500	0.0263	0.9345	168
800–100–100–2	90,402	500	0.0268	0.9639	114
800–200–200–2	200,802	500	0.0267	0.9594	189
800–200–200–200–2	241,002	500	0.0377	0.9589	213
800–200–100–20–2	182,362	500	0.0377	0.9558	179

time-domain wave signals and one-dimensional convolutional neural network (1-D CNN), the important temporal information of the original Lamb wave signal was preserved, which allows accurate localization of damages in plates with few transducers. An inverse problem in nonlinear elastostatics was considered by Stavroulakis et al. [21], in which concerns the identification of unilateral contact cracks by means of boundary displacement measurements for given static loadings. Chen et al. [22] developed a novel deep learning inverse solution of identification method to determine and identify the impact load conditions of shell structures based on their final state of damage or inelastic deformation.

Most of the inverse problems in the literature as mentioned above are solved by the classification techniques [23]. By giving several reasonable solutions in advance, the solved features satisfy the conditions for one of the solutions, then which is considered as the real solution to the problems, although this solution may not be the best of all the solutions to this problem. However, a practical issue for a category of inverse problems is that the dataset is not suitable to be divided into classes, and it makes the classification approaches unfeasible. Therefore, for problems involving accurate quantitative predictions, such as defect

localization, which will be investigated in this work, the regression approach will be more reliable.

In this work, we propose the combination of the ML and the boundary element method (BEM) to address two major issues in regression method-based data-driven mechanics problems: (i) the accuracy of the datasets directly affects the performance of the network; (ii) large number of datasets is needed for the networks to extract the input-output mappings. Based on analytical fundamental solutions, the BEM based on the boundary integral equations can deliver high accuracy. The discretization process is only carried out on the boundaries where the solutions are needed as the input, and therefore, it immediately reduces the computation work load on data generation. An effective supervised machine learning model based on regression algorithms is constructed by capturing the relationship between the input boundary strains and the output defect location. The flexible parameters are adjusted to minimize a cost function. Exact predictions of the defect locations are then generated using the test dataset.

## 2. Methodology

The inverse problem of defect localization is solved using single or multi-layer perceptron [2]. For this specific problem, the networks require less than three fully-connected hidden layers with 2 to 200 neurons, in addition to the input and output layers. Boundary strains ( $\epsilon_x$  and  $\epsilon_y$ ) obtained from the BEM are given as the input neurons, which are used to predict the center coordinates ( $c_x$  and  $c_y$ ) and radius ( $r$ ) of the unknown circular defect inside the plate, as shown in Fig. 1.

### 2.1. Dataset preparation

A two-dimensional (2D) plate with a circular defect is fixed on the left boundary and subject to a tensile pressure  $p$  on the right boundary as shown in Fig. 2(a). In computational mechanics, we usually calculate the stress and strain distribution based on the known defect location and boundary conditions. In contrast, for the inverse problem considered in this paper, the boundary conditions are unchanged but the defect location is varied. A FORTRAN program of the BEM for 2D elasticity problems with constant elements [24] is employed to obtain the strain distribution on the boundary nodes of this defected plate as the input dataset. The center coordinates and the radius of the circular defect are collected as the output dataset.

The equilibrium and stress-strain relations are described as

$$\sigma_{ij,j} = 0, \text{ and } \epsilon_{ij} = \frac{1}{E} [(1 + \nu)\sigma_{ij} - \nu\sigma_{kk}\delta_{ij}], \quad (1)$$

respectively, in which  $\sigma_{ij}$  and  $\epsilon_{ij}$  are the stress and strain tensors,

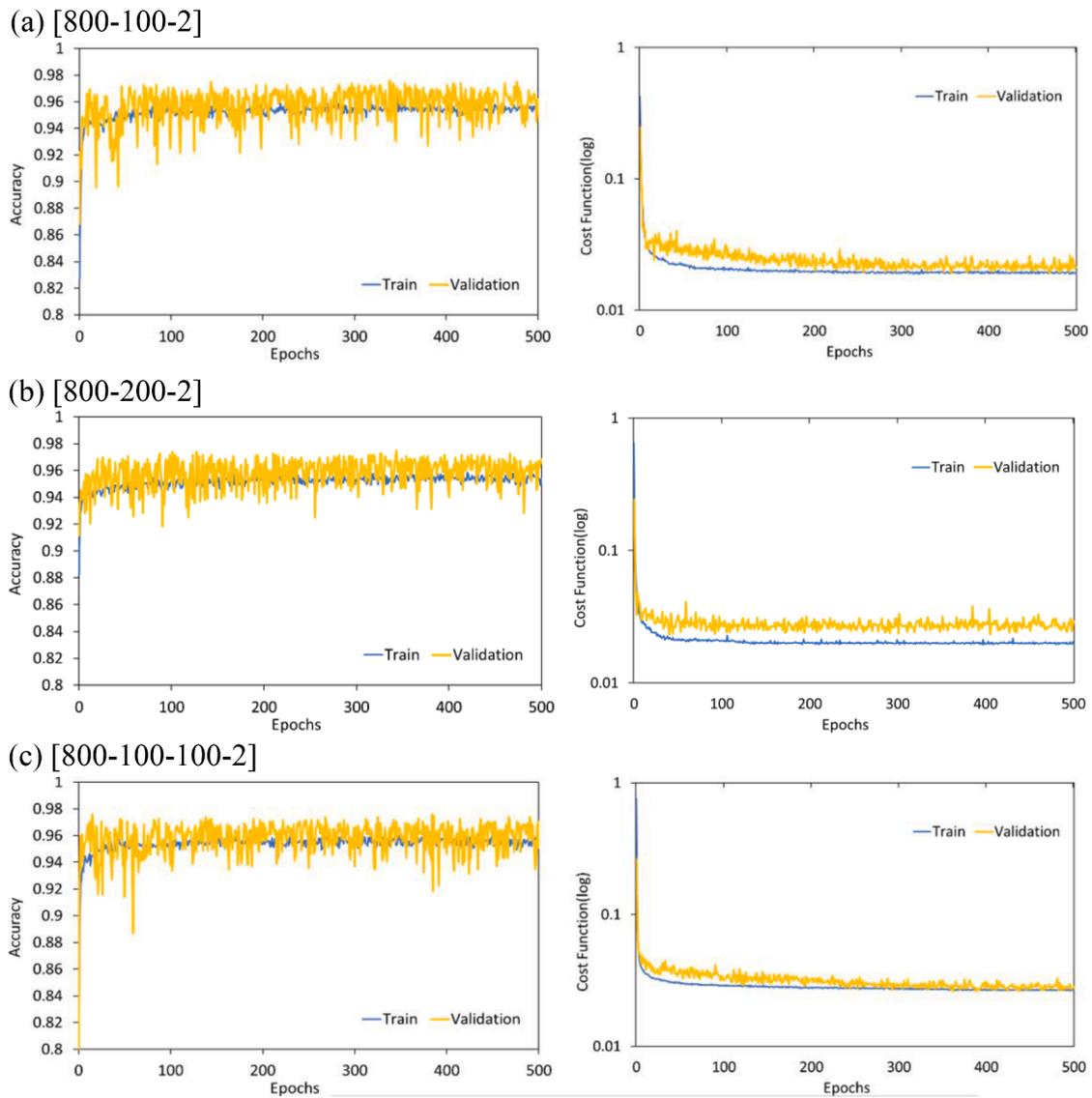


Fig. 5. Accuracy and cost function training history of case 1.

respectively, and  $\delta_{ij}$  is the Kronecker delta symbol. Since the BEM can reduce problem dimension by one [24–26], only boundary elements on the edges of the plate and the circular are needed. Each edge of the plate is discretized by 100 constant elements and the edge of the circular is discretized by 360 constant elements as plotted in Fig. 2(b). In this study, the Young’s modulus  $E = 1N/m^2$ , Poisson’s ratio  $\nu = 0.3$  and the pressure load  $p = 1Pa$ . For verification, the stresses of four points located on the edge of the circular pointed in Fig. 2 are compared with the analytical results and listed in Table 1. The boundary strains of the model without circular defect are also compared with the analytical results, the location of the four chosen points and the corresponding BEM model are illustrated in Fig. 3 and the errors are almost 0 (Table 2).

2.2. Data normalization

The collected data from the previous step have a wide range of the data types and scales which raise the difficulties in feature identification, and therefore, it is not suitable to be directly used for model training. Thus, data transformation is used to normalize the input data to meet the requirements of machine learning algorithms [27–29].

In this research, the Z-score normalization [27–28], which is one of

the most popular normalization schemes, is employed to transform the data. Assume a sample with  $N$  data,  $\{x^n\}^N$ , its mean value and variance are 0 and 1, respectively, and given as:

$$\mu = \frac{1}{N} \sum_{n=1}^N x^n \text{ and } \sigma_{variance}^2 = \frac{1}{N} \sum_{n=1}^N (x^n - \mu)^2 \tag{2}$$

The set of data after the Z-score normalization yields:

$$\hat{x}^n = \frac{x^n - \mu}{\sigma_{variance}} \tag{3}$$

After the data normalization, the total data is split into two sets, one for training the network, and the other for testing. In the present paper, 70% data are used for training the neural network and the remaining 30% data are used for testing.

2.3. Neural network optimization

Neural network optimization is a process designed to achieve a high-quality network structure, as well as to improve the generalization ability of the model. In the present work, mini-batch gradient descent algorithm [27–28] is applied to optimize the network structure. The

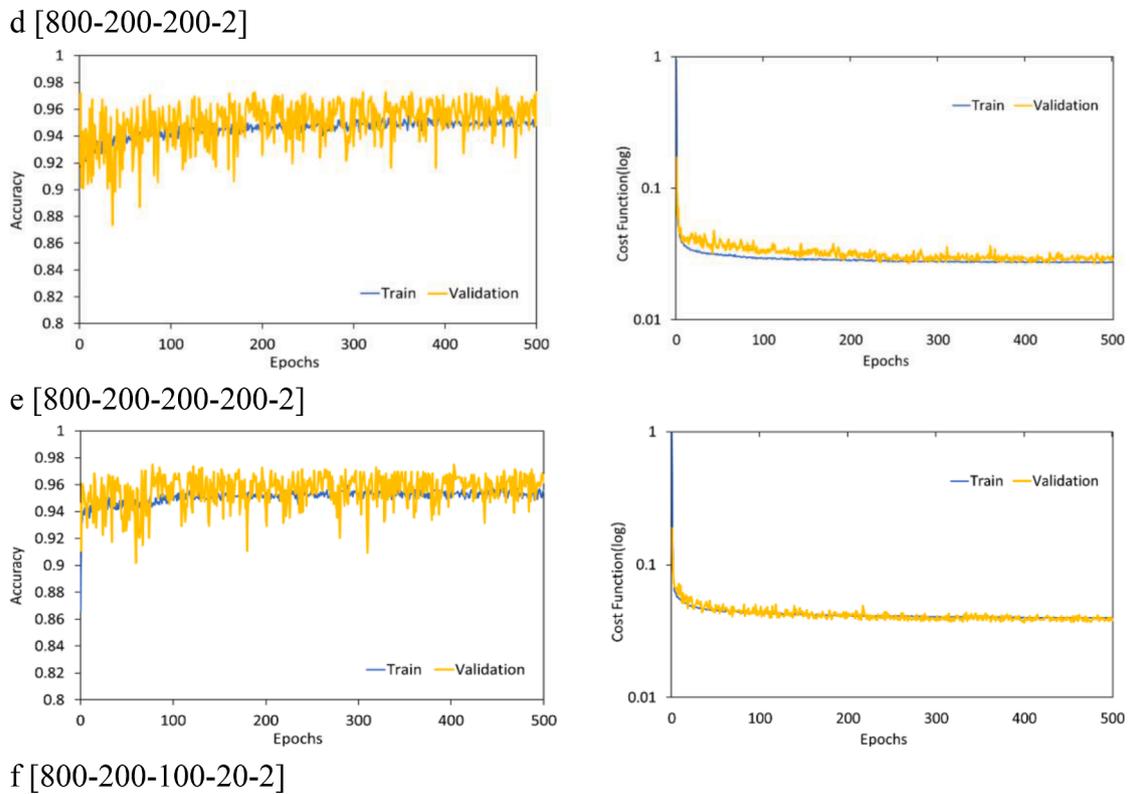


Fig. 5. (continued).

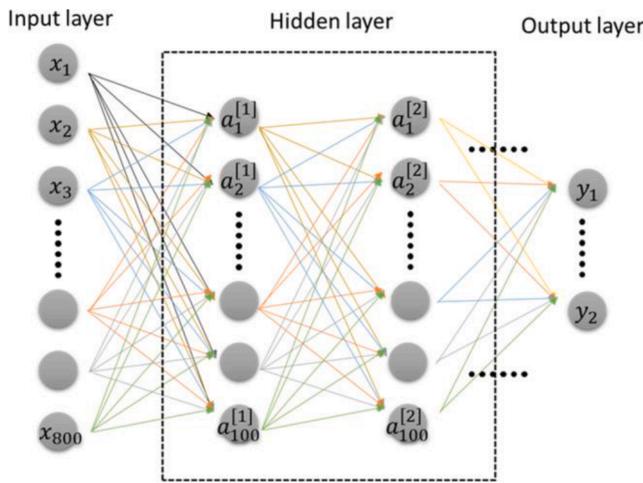


Fig. 6. Architecture of the prediction model for case 1.

training set is divided into several small batches to compute the loss function and the gradient one batch after another, and then the corresponding parameters are updated. The mini-batch gradient descent algorithm improves the training efficiency and reduces the possibility of overfitting. The proposed neural network is constructed in sequence by one input layer,  $L$  hidden layers and one output layer, where each hidden layer contains  $K$  neurons, and the input and output layer have  $N$  and  $J$  neurons, respectively. Eq. (4) shows the general structure of the neural network:

$$O_j = f_{activation} \left( \sum_{m=1}^{K^{L_j}} W_m^{L_j} T_m^{L_j-1} + B_m^{L_j} \right), j = 1, 2, 3 \quad (4)$$

where

$$f_{activation} = ReLU = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (5)$$

which is a non-linear activation functions for the output layer and hidden layers. The ReLU (Rectified Linear Unit) function [27–28] is employed to active almost 50% neurons, which makes the neural network sparse. The fact that the derivative of ReLU being equal to 1 at  $x \geq 0$  avoids gradient explosion during training and improves the convergence efficiency. The parameters to be trained are weights,  $W$  and biases,  $B$ . The input layer consists of  $N = 2n$  input neurons, each of which represents the strain value of either  $\varepsilon_x$  or  $\varepsilon_y$  for total  $n$  boundary nodes, whereas the output layer consists of  $J = x003D3$  neurons, each of which represents center coordinates of the circular defect and its radius. Weights and biases are tuned using the root mean square prop (RMSprop) [27–28] algorithm.

The purpose of the learning or training process in the ANN methodology is to reduce the error at the output layer. This is done by adjusting the weight and bias of each neuron through back-propagation algorithm. The error of the output layer is described by a loss function,  $f_{loss}(\text{training})$ , which is the deviation between the target value versus the predicted value. In this paper, the mean absolute error (MAE) [27–28] is taken as the loss function for the training process:

$$f_{loss} = \frac{1}{J} \sum_{j=1}^J [y_{jTarget} - y_{jPred}], J = 2 \text{ or } 3 \quad (6)$$

Due to the powerful data fitting ability of ML, the fine-tuned weights and biases will result in very small loss function values, but this often leads to overfitting and impedes generalization for the unseen datasets. Therefore, regularization [27–28] is employed to avoid overfitting by limiting model complexity. Here,  $\ell_2$  norm is used to regularize the neural network, and the loss function can be rewritten as:

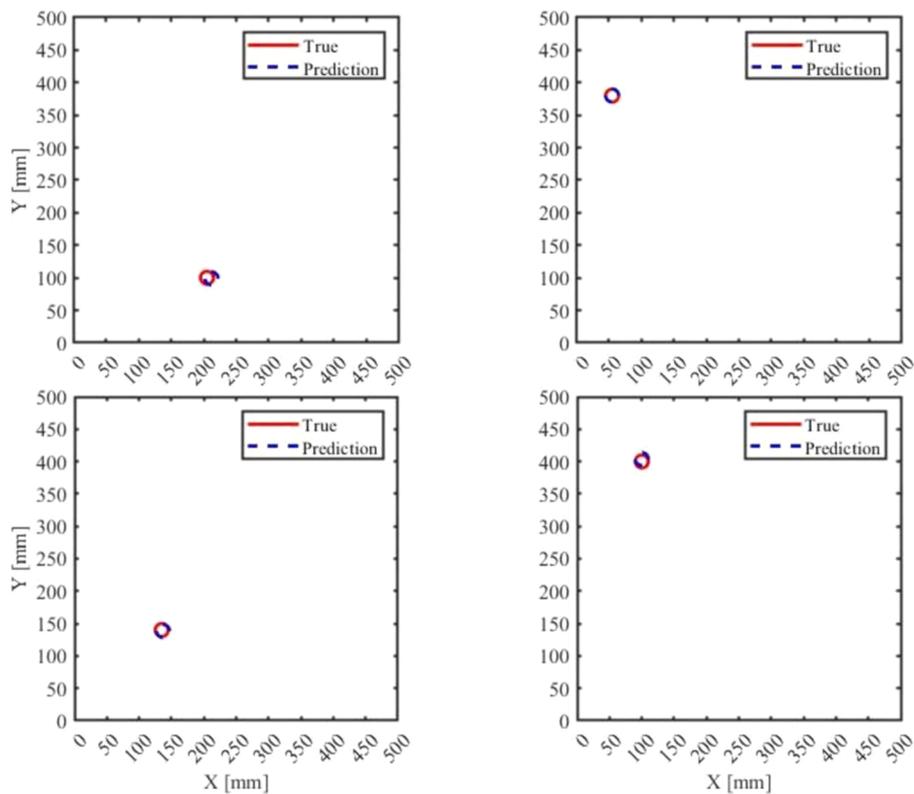


Fig. 7. Comparison of the predictions and truth detect locations for case 1.

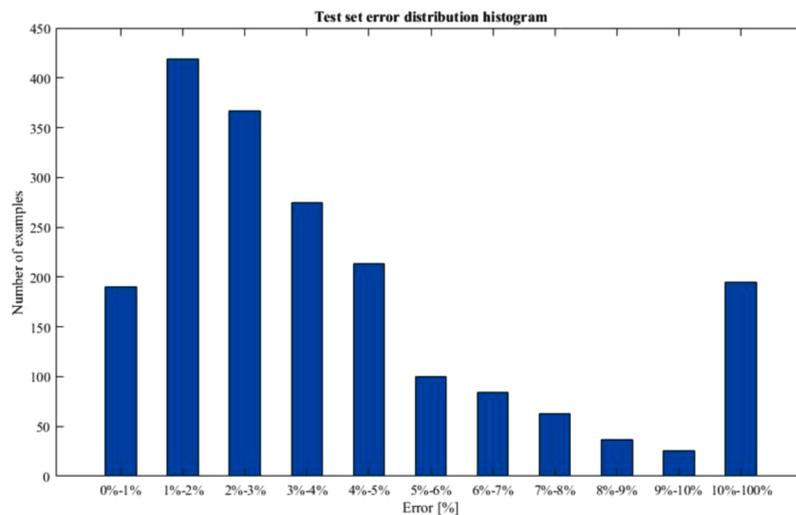


Fig. 8. Test set error distribution histogram for case 1.

$$f'_{loss} = \frac{1}{J} \sum_{j=1}^J [y_{jTarget} - y_{jPred}] + \lambda \ell_2 \tag{7}$$

where,  $\lambda = 0.01$  is the coefficient of  $\ell_2$  norm.

### 3. Results and discussion

In this section, a steel plate with Young’s modulus  $E = 2.06 \times 10^8 N/m^2$ , Poisson’s ratio  $\nu = 0.28$  and the tensile pressure  $p = 1000Pa$  is considered.

#### 3.1. Case 1: Square plate with a circular defect

A square plate of side length 500 mm with a circular defect of fixed radius 10 mm is plotted in Fig. 4(a). Each boundary and the perimeter of the plate is discretized into 100 and 360 elements, respectively, as illustrated in Fig. 4(b). Data are collected by moving the center of the circular along x and y axis 1 mm at a time, and each sample contains the strain distribution on the boundaries as the input and the coordinates of the defect center as the output. 6561 samples are collected in total for the current plate configuration.

The configuration of NN models is defined in Table 3. Six NN models with different number of layers and neurons are tested and the results are shown in Table 4. All networks exhibit high accuracy between

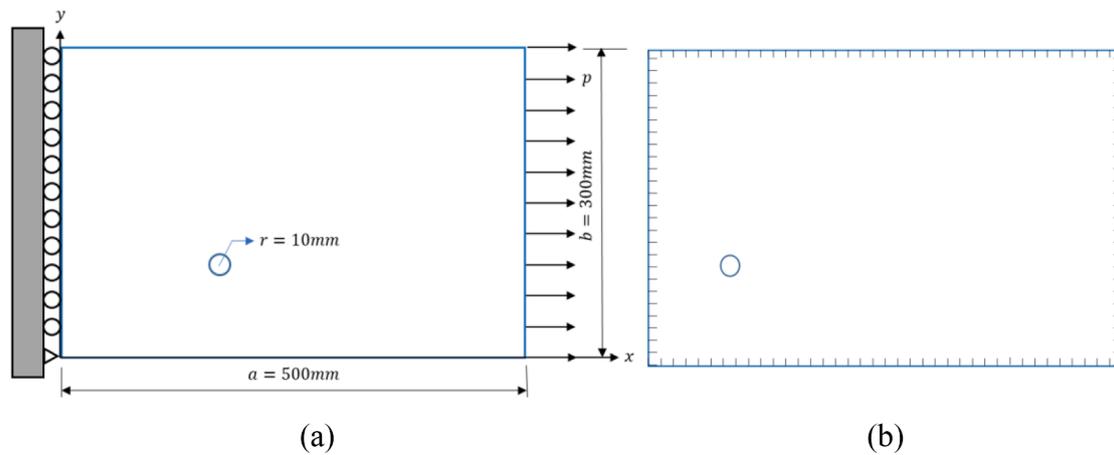


Fig. 9. (a) Geometry and (b) BEM model of the studied rectangular plate with a circular defect.

**Table 5**  
ANN parameters for case 2.

Number of examples in Training dataset	2268
Number of examples in Testing dataset	972
Batch-size	32
Number of neurons in the input layer	800
Number of hidden layers	1–3
Number of neurons in the hidden layers	20–200
Number of neurons in the output layer	2
Activation layers	ReLU
Number of epochs	500
Validation split	0.2
Optimizer	RMSprop

**Table 6**  
Model performance with different architectures for case 2.

Neurons at each layer	Trainable parameters	Number of Epochs	Loss value (m)	Accuracy	CPU time (s)
800–100–2	80,302	500	0.0160	0.9660	234
800–100–100–2	90,402	500	0.0204	0.9671	238
800–200–100–2	180,502	500	0.0208	0.9455	301
800–200–100–20–2	182,362	500	0.0296	0.9722	311

93.45% and 96.39% at predicting of center coordinates. The 800–100–2 configuration reaches an accuracy of 95.12% after 500 epochs with 80,302 trainable parameters. A less notable difference in the accuracy with increasing the number of trainable parameters, but an exponential time cost increased with increasing the hidden layer of the neural network architectural.

Fig. 5 shows the training and validation history of the accuracy and cost function. There is a good agreement between the training and validation accuracy. Loss values in the considered models ranges from 0.0190 to 0.0377, after a stabilization period and stay roughly consistent with increasing number of epochs. Since the NN configuration of 800–100–100–2 depicted in Fig. 6 shows the highest accuracy with relatively small number of trainable parameters, which gives much higher efficiency, we choose this configuration for testing.

A visualization of localization results, which is obtained from random testing data, is given in Fig. 7. The predictions match very well with the true locations.

Root mean square errors of the test dataset are analyzed and the corresponding histogram presented in Fig. 8. Almost 74% predicted results can achieve errors less than 5%. However, there are about 200 samples, which comprise only 9% of the total samples, result in higher errors over 10%. This is believed to be an intrinsic difficulty of the

regression network compared to the classifying network for the inverse problem, for the latter depends less on the accuracy of the numerical values. Indeed, the higher error level can be reduced by increasing the volume of training dataset, but it cannot be completely avoided. This behavior of the regression network will be verified by the following examples.

### 3.2. Case 2: Rectangular plate with a circular defect

In this case, a different plate geometry is investigated. A rectangular steel plate with the same material properties, boundary supports and tensile pressure distribution as in case 1 is considered. The length and the width of the plate is 500 mm and 300 mm, respectively and a circular defect with a fixed radius 10 mm is placed inside the plate. Each boundary of the plate and the perimeter of the circular is again discretized into 100 and 360 elements, respectively. The detail geometry and the mesh description of the plate is illustrated in Fig. 9.

Data are collected by moving the center of the circular along x and y axis in 1 mm spacing at a time, and each sample contains the strain distribution on the boundaries as the input and the coordinates of the defect center as the output. 3240 samples are collected in total for the current plate configuration. After the data normalization, 70% of the dataset are used for training the model and the other 30% are used for prediction. The configuration of NN models tested in case 2 is defined in Table 5. Four NN models with different number of layers and neurons are tested and the results are shown in Table 6. All networks exhibit high accuracy. Three of the networks achieve the accuracy over 96%, and the highest accuracy is 97.22%. Fig. 10. shows the training and validation history of the accuracy and cost function.

Loss values in the considered modes ranges from around 0.0160 to 0.0296, after an initial stabilization period. As indicated in Fig. 10, the loss values stay roughly consistent with increasing number of epochs after a stabilization period. It also can be seen that there is a good agreement between the training and validation accuracy. The NN configuration of 800–200–100–20–2 shows the highest accuracy and is chosen to predict the results as shown in Fig. 11.

A visualization of localization results, which is obtained from random testing data for case 2, is given in Fig. 12. It can be seen that, the predicted defect locations match very well with the truth locations.

Root mean square errors of the test dataset are analyzed and the corresponding histogram of case 2 are presented in Fig. 13. For this case, almost 64% predicted results can achieve errors less than 5%. However, about 130 samples, which comprise only 13% of the total samples result a higher error over 10%.

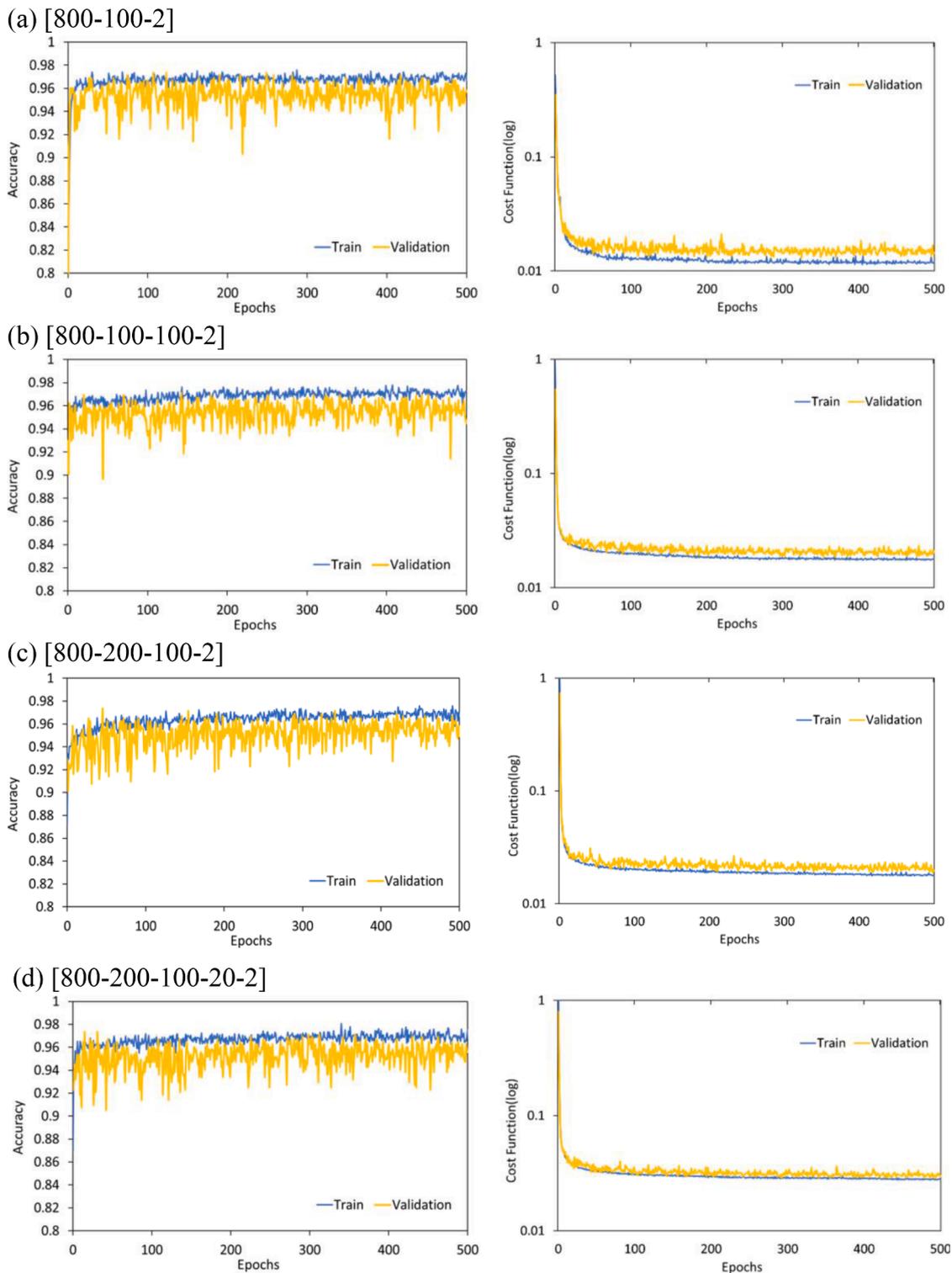


Fig. 10. Accuracy and cost function training history of case 2.

### 3.3. Case 3: Rectangular plate with a tiny circular defect

In case 3, a same rectangular steel plate with the same material properties, boundary supports and tensile pressure distribution as in case 2 is considered. Only the radius of the circular defect in the plate is reduced to 2 mm. Data are also collected by moving the center of the circular defect along x and y axis 1 mm at a time, and each sample contains the strain distribution on the boundaries as the input and the coordinates of the center as the output. 3240 samples are collected in

total for the current plate configuration. Among of these data, 70% are used for training and the other 30% are used for prediction. Only one NN architecture was constructed and tested in case 3 which is defined in Table 7. The results for this NN configuration are shown in

Table 8. The accuracy and the cost function in log form of this NN configuration are plotted in Fig. 14.

A high accuracy of 97.53% and loss value 0.0159 can be achieved by the current constructed NN after an initial stabilization period. As the same, the loss values stay roughly consistent with increasing number of

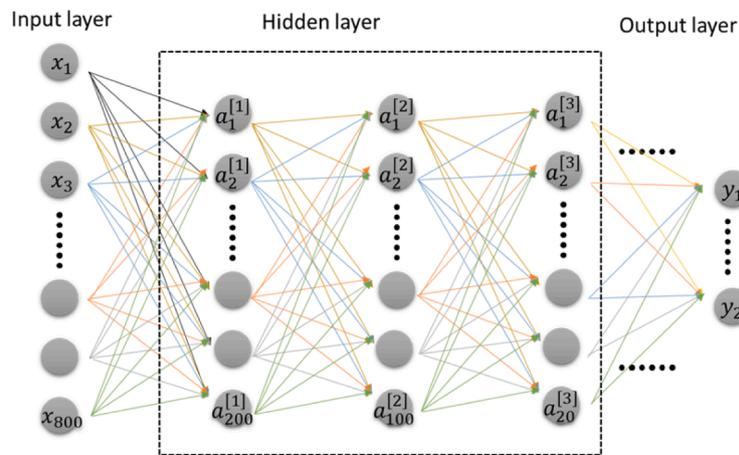


Fig. 11. Architecture of the prediction model for case 2.

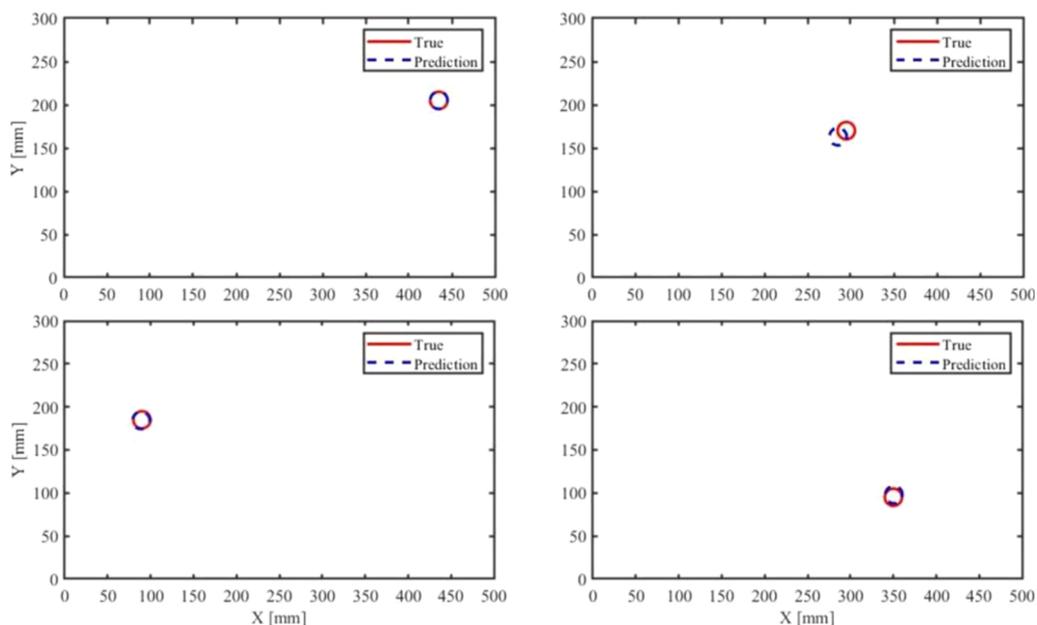


Fig. 12. Comparison of the predictions and truth defect locations for case 2.

epochs as shown in Fig. 14. Comparing the behavior of the validation dataset against the training one, good performance is observed by the current NN. The NN configuration of 800–200–100–20–2 presented in Fig. 15 was taking to predict the results.

A visualization of localization results, which is obtained from random testing data for case 3, is given in Fig. 16. It can be seen that the predicted defect locations match very well with the truth locations.

Root mean square errors of the test dataset are analyzed and the corresponding histogram of case 3 are presented in Fig. 17. For this case, the prediction accuracy is high, almost 80% predicted results can achieve the error less than 5%. 44 samples, which comprise only 4.5% of the total samples result in a higher error over 10%.

### 3.4. Case 4: Rectangular plates with various radius of circular defect

Based on case 3, the various radius of the circular defect ranging from 7 mm to 20 mm in the rectangular plate is extracted as another feature in the case 4 model analysis. Data are collected not only by moving the center along x and y axis 1 mm at a time, also by increasing the radius of the circular defect 1 mm at a time. Each sample contains the strain distribution on the boundaries as the input and the center

coordinates as well as radius of the circular defect as the output. 45,360 samples are collected in total for the current plate configuration. The number of the dataset are almost 8–10 times than former cases. All the data normalized by Z-score normalization and 70% of them are used for training the model. The configuration of NN modes tested in case 4 is defined in Table 9. Two NN models with different number of layers and neurons are tested and the results are shown in Table 10. The accuracy and the cost function in log form of these NN configurations are plotted in Fig. 18.

All these two NNs can achieve a higher accuracy between 98.18% and 98.64% at the prediction of center coordinates and the radius of the circular defect in the rectangular plate after 500 epochs. The accuracy is much higher than the former cases, which is due to the large number of the training dataset. Loss values in the considered models ranges from 0.7061 to 0.7275, after an initial stabilization period. As indicated in Fig. 18, the loss values stay roughly consistent with increasing number of epochs. Comparing the behavior of the validation dataset against the training one, good performance is observed by the predictions of the NNs. The NN configuration of 800–100–3 (Fig. 19) reaches an accuracy of 98.64% after 500 epochs with 80,403 trainable parameters is applied to predict the results.

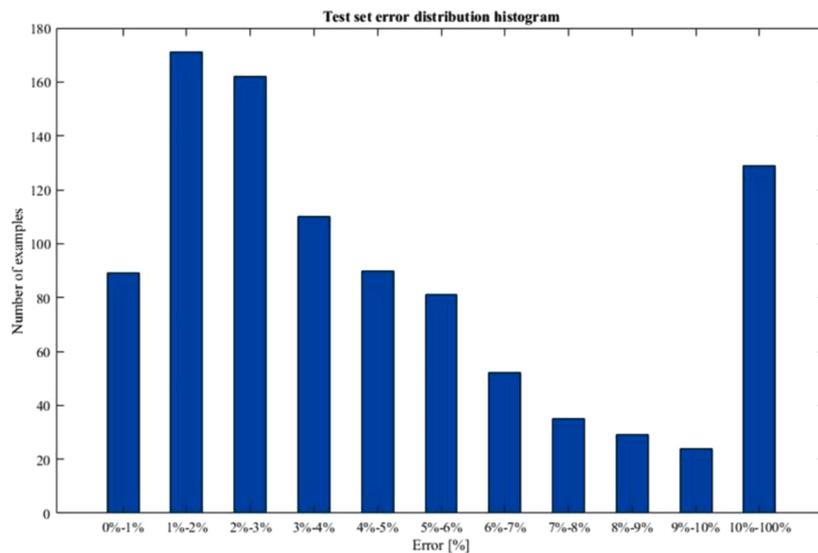


Fig. 13. Test set error distribution histogram for case 2.

Table 7

ANN parameters for case 3.

Number of examples in Training dataset	2268
Number of examples in Testing dataset	972
Batch-size	32
Number of neurons in the input layer	800
Number of hidden layers	3
Number of neurons in the hidden layer	20–200
Number of neurons in the output layer	2
Activation layers	ReLU
Number of epochs	500
Validation split	0.2
Optimizer	RMSprop

Table 8

Model performance with different architectures for case 3.

Neurons at each layer	Trainable parameters	Number of Epochs	Loss value (m)	Accuracy	CPU time (s)
800–200–100–20–2	182,362	500	0.0159	0.9753	80

A visualization of localization results, which is obtained from random testing data for case 4, is given in Fig. 20. It can be concluded that, the predicted locations and the size of the defects matched very well with the truth defects.

Root mean square errors of the test dataset are analyzed and the corresponding histogram of case 4 are presented in Fig. 21. For this case, almost 85% predicted results can achieve error less than 5%, and most of the predicted results errors are distributed between 0% and 2%. With an increasing number of training data, the prediction accuracy of the model is greatly improved, and it is reflected by the rapidly declined volume of the samples with over 6% error, which results from only 30 samples, comprising only 0.22% of the total test samples.

### 3.5. Discussions

In this work, identifications of defect size and localization are realized based on the combination of the BEM and multilayer perceptron. The performance of the proposed algorithm is evaluated systematically at four generalization levers:

- Testing on the square plate with different defect locations from the training data.
- Testing on the rectangular plate with different defect locations from the training data.
- Testing on the rectangular plate with tiny defect at different locations from the training data.
- Testing on the rectangular plate with various size and location of defect based on the training data.

For all of these cases, only boundary strains of the considered structures are needed as the input, which can be directly collected by the

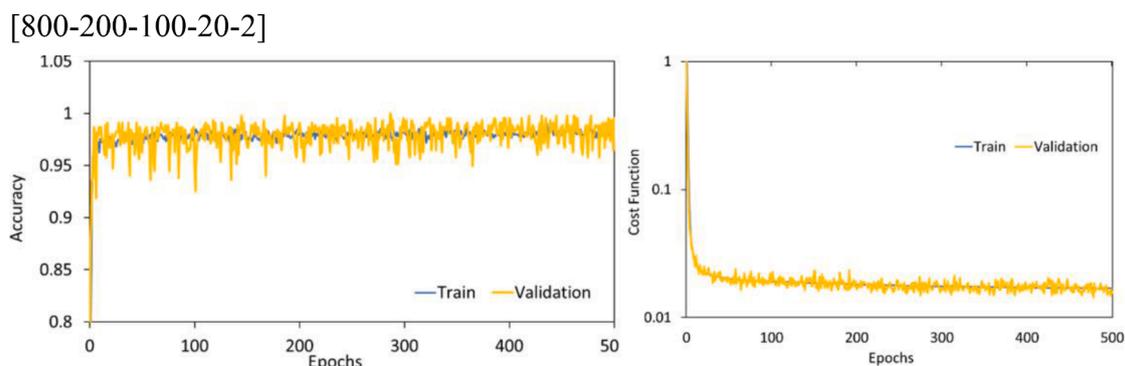


Fig. 14. Accuracy and cost function training history of case 3.

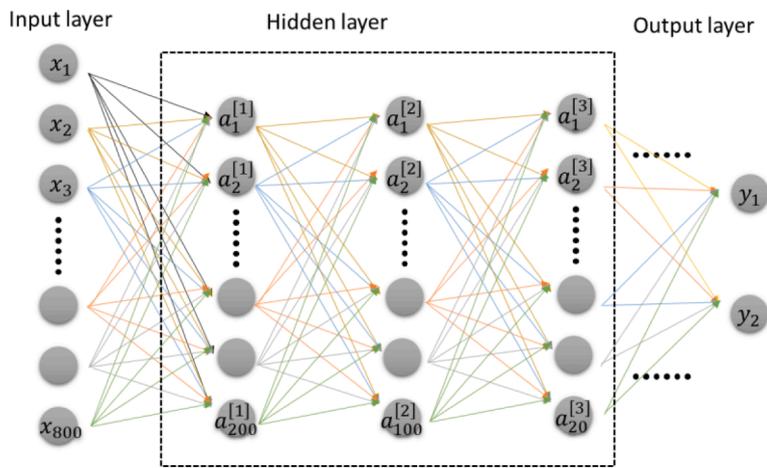


Fig. 15. Architecture of the prediction model for case 3.

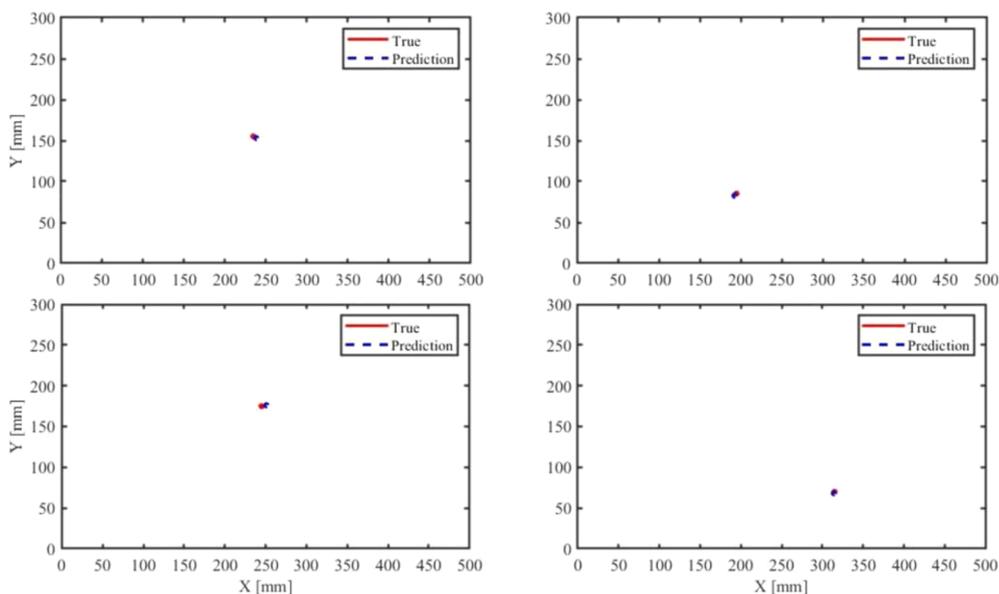


Fig. 16. Comparison of the predictions and truth detect locations for case 3.

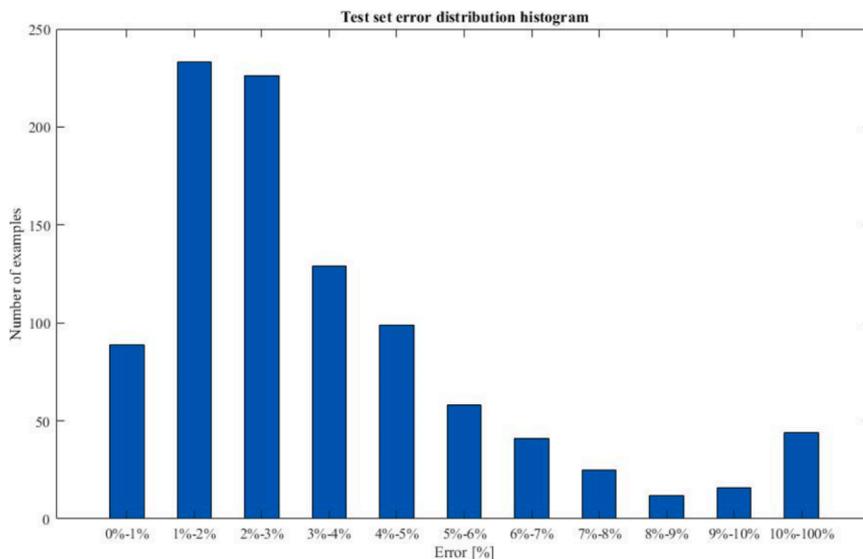


Fig. 17. Test set error distribution histogram for case 3.

**Table 9**

ANN parameters for case 4.

Number of examples in Training dataset	31,751
Number of examples in Testing dataset	13,609
Batch-size	32
Number of neurons in the input layer	800
Number of hidden layers	1–2
Number of neurons in the hidden layer	100
Number of neurons in the output layer	3
Activation layers	ReLU
Number of epochs	500
Validation split	0.2
Optimizer	RMSprop

**Table 10**

Model performance with different architectures for case 4.

Neurons at each layer	Trainable parameters	Number of Epochs	Loss value (cm)	Accuracy	CPU time (s)
800–100–100–3	90,530	500	0.7061	0.9818	701
800–100–3	80,403	500	0.7275	0.9864	628

using the strain gauges in the real applications. Single or multilayers fully connected neural networks are constructed to train the data and predict the results. The process is simple and reduces the errors in the data transmission. In addition, a high lever accuracy of about 98% can be achieved.

In the practical application, most of the structures have fixed geometry, such as beams and columns in civil engineering, as well as machine components in mechanical engineering. The dataset of boundary strains is easily obtained by using the strain gauges on the boundaries or analyzed by the BEM. Training an appropriate neural network from the dataset can accurately predict the defect information of the structures. Thus, the proposed algorithm has a great potential in

practical applications.

#### 4. Conclusions

In this paper, a machine learning computational mechanics method based on fully connected multi-layer perceptron is proposed to solve for 2-D inverse elasticity problems. Accurate prediction of the location of a circular defect and its radius in a plate is achieved by examining the boundary strain values. The BEM is first applied to accurately calculate a large amount of boundary strain values corresponding to various plate geometries and circular defect locations and radii. The strain values

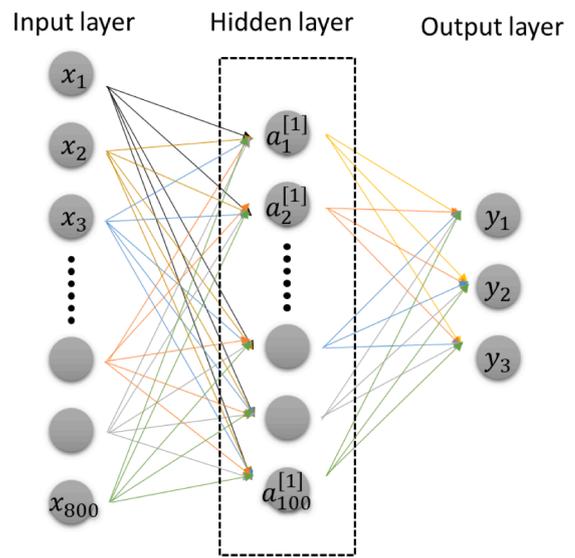
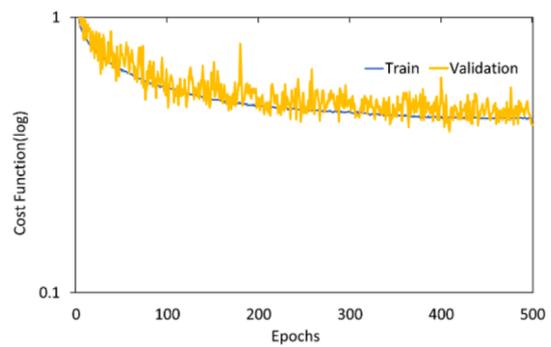
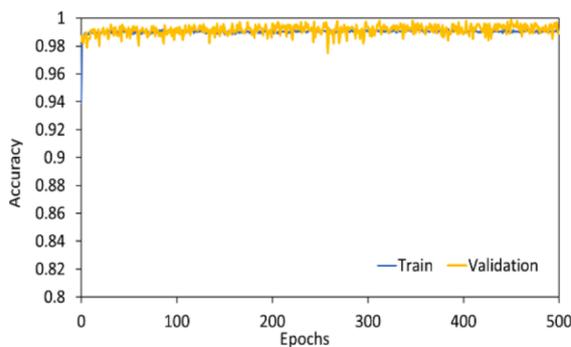


Fig. 19. Architecture of the prediction model for case 4.

(a) [800-100-100-3]



(b) [800-100-3]

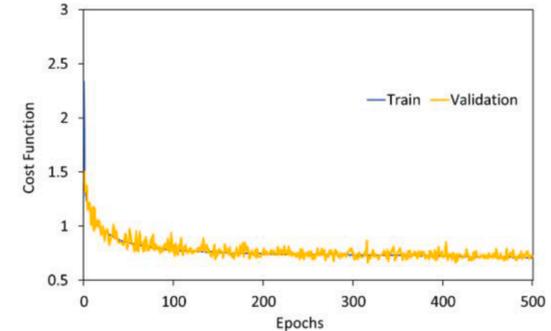
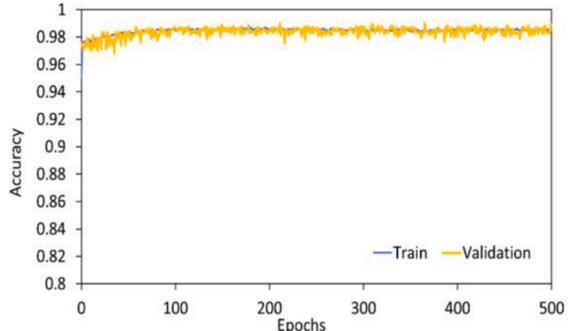


Fig. 18. Accuracy and cost function training history of case 4.

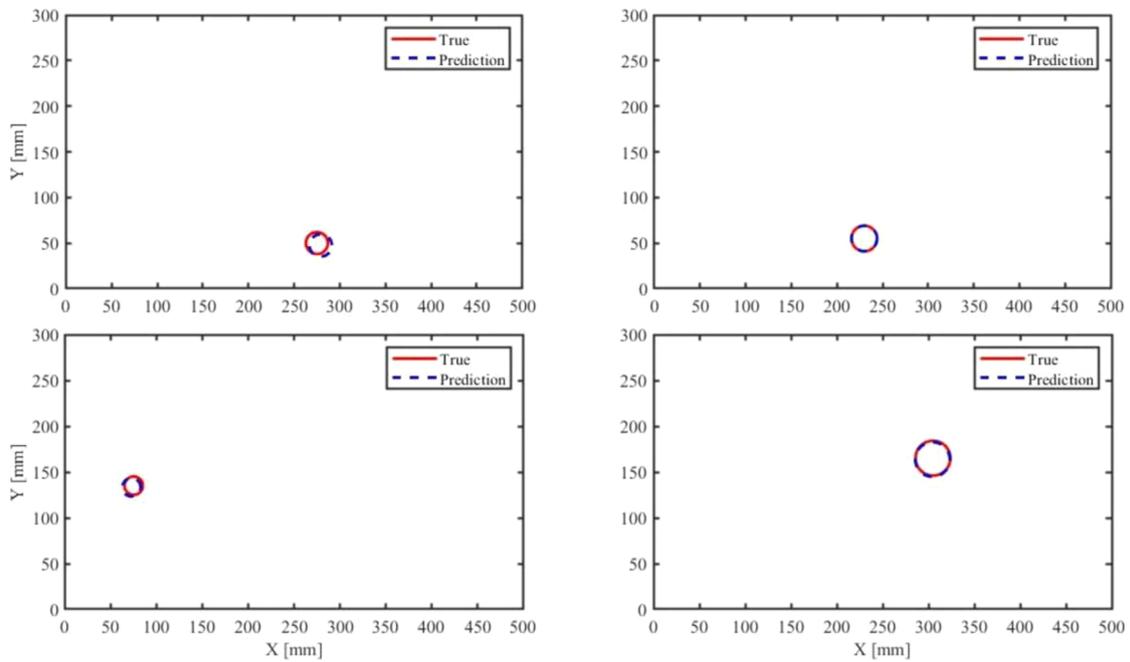


Fig. 20. Comparison of the predictions and truth defect for case 4.

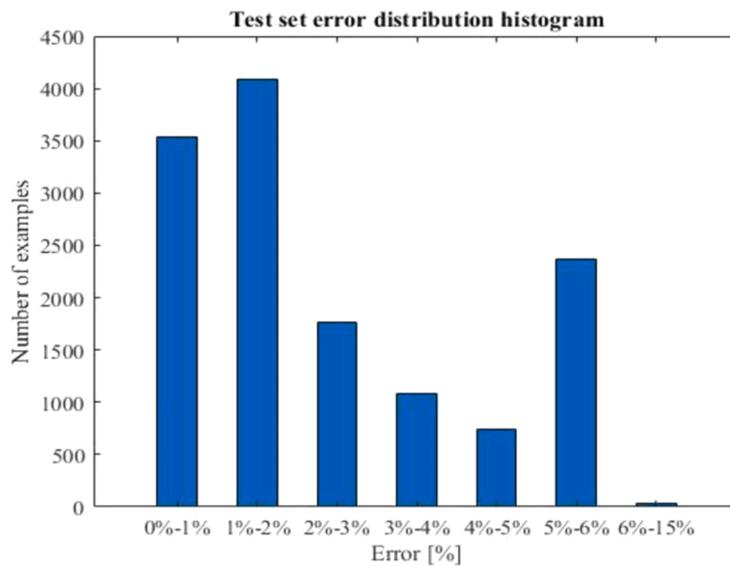


Fig. 21. Test set error distribution histogram for case 4.

along with the center coordinates and radii are divided as training, validation, and testing datasets, where the training set were then normalized using the Z-Score method before ML. ReLU function is used as the activation function of the neural network, and the model is trained iteratively using the RMSProp algorithm as the optimizer. Regularization based on  $l_2$  norm is used to prevent the training process from overfitting. The results show that the predicted results can achieve about 98% accuracy. The proposed algorithm provides a new idea for solving inverse problems in computational mechanics. In additional, compared with the Lamb wave based SHM techniques, the proposed approach only requires the boundary strains of the considered structures as the input data and a simple training process for the regression problem. Therefore, it is much easier to implement and it has the potential in real SHM applications.

**Declaration of Competing Interest**

None.

**Acknowledgements**

The authors would like to thank the support from the National Natural Science Foundation of China (Project Nos. 11901283 and 11972179), the Natural Science Foundation of Guangdong Province (No. 2020A1515010685), the Department of Education of Guangdong Province (No. 2020ZDZX2008), and Natural Science Foundation of Shaanxi Province (Project No. 2018JQ5079).

## References

- [1] Adler J, Öktem O. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Prob* 2017;33:124007.
- [2] Kirchdoerfer T, Ortiz M. Data-driven computational mechanics. *Comput Methods Appl Mech Eng* 2016;304:81–101.
- [3] Nguyen L, Keip MA. A data-driven approach to nonlinear elasticity. *Comput Struct* 2018;194(1):97–115.
- [4] Mallela UK, Upadhyay A. Buckling load prediction of laminated composite stiffened panels subjected to in-plane shear using artificial neural networks. *Thin Walled Struct* 2016;102(5):158–64.
- [5] Capuano G, Rimoli JJ. Smart finite elements: a novel machine learning application. *Comput Meth Appl Mech Eng* 2019;345:363–81.
- [6] Liang L, Liu ML, Martin C, Sun W. A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. *J R Soc, Interface* 2018;20170844.
- [7] Haghghat E, SciANN JR. A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Comput Meth Appl Mech Eng* 2021;373:113552.
- [8] Guo YF, Wang CR, Ma Z, Huang XH, Sun KW, Zhao RL. A new mesh smoothing method based on a neural network. *Comput Mech* 2021;196:1–14.
- [9] Gorji MB, Pannemaecker A, Spevack S. Machine learning predicts fretting and fatigue key mechanical properties. *Int J Mech Sci* 2022;215:106949.
- [10] Finol D, Lu Y, Mahadevan V, Srivastava A. Deep convolutional neural networks for eigenvalue problems in mechanics. *Int J Numer Methods Eng* 2019;118(5):258–75.
- [11] Wang P, Shao YC, Wang HT, Yang W. Accurate interatomic force field for molecular dynamics simulation by hybridizing classical and machine learning potentials. *Extreme Mech Lett* 2018;24:1–5.
- [12] Li X, Yan ZM, Liu ZL. Combination and application of machine learning and computational mechanics, 64. Science China Press; 2019. p. 635–48.
- [13] HR Tamaddon-Jahromi, Chakshu NK, Sazonov I, Evans LM, Thomas H, Nithiarasu P. Data-driven inverse modelling through neural network (deep learning) and computational heat transfer. *Comput Meth Appl Mech Eng* 2020;369:113217.
- [14] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 2019;378:686–707.
- [15] Tang MJ, Tang HS. A physics-informed deep learning method for solving forward and inverse mechanics problems of thin rectangular plates. *Chinese J Computat Mech* 2020:1–9.
- [16] He ZL, Ni FT, Wang WG, Zhang J. A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials. *Mater Today* 2021;28:102719.
- [17] Dwivedi V, Parashar N, Srinivasan B. Distributed learning machines for solving forward and inverse problems in partial differential equations. *Neurocomputing* 2020;420(5):299–316.
- [18] Dworakowski Z, Dragan K, Stepinski T. Artificial neural network ensembles for fatigue damage detection in aircraft. *J Intell Mater Syst Struct* 2017;28:851–61.
- [19] De Fenza A, Sorrentino A, Vitiello P. Application of artificial neural networks and probability ellipse methods for damage detection using lamb waves. *Compos Struct* 2015;133:390–403.
- [20] Zhang SY, Li CM, Ye WJ. Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network. *Mech Syst Sig Process* 2021;147:107107.
- [21] Stavroulakis GE, Antes H. Nondestructive elastostatic identification of unilateral cracks through BEM and neural networks. *Comput Mech* 1997;20:439–51.
- [22] Chen GR, Li TG, Chen QJ, Ren SF, Wang C, Li SF. Application of deep learning neural network to identify collision load conditions based on permanent plastic deformation of shell structures. *Comput Mech* 2019;64:435–49.
- [23] Müller A, Karathanasopoulos N, Roth CC, Mohr D. Machine Learning Classifiers for Surface Crack Detection in Fracture Experiments. *Int J Mech Sci* 2021;209:106698.
- [24] Liu YJ. Fast multipole boundary element method: theory and applications in engineering. Cambridge: Cambridge University Press; 2009.
- [25] Yang Y, Liu YJ. A new boundary element method for modeling wave propagation in functionally graded materials. *Eur J Mech A Solids* 2020;80:103987.
- [26] Yang Y, Kou KP, Lam CC, Iu VP. Dynamic behaviors of tapered bi-directional functionally graded beams with various boundary conditions under action of a moving harmonic load. *Eng Anal Boundary Elem* 2019;104:225–39.
- [27] Aggarwal CC. Training deep neural networks. *Neural Networks and Deep Learning. A Textbook*. 2018. p. 105–67. <https://doi.org/10.1007/978-3-319-94463-0> (Chapter 3).
- [28] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.
- [29] Chollet F. Deep learning with python. Manning; 2017.