A damage-informed neural network framework for structural damage identification

Hau T. Mai[b,1], Seunghye Lee[a,2], Joowon Kang[c], Jaehong Lee[a,*]

[a]*Deep Learning Architectural Research Center, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Republic of Korea*
[b]*Faculty of Mechanical Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam*
[c]*School of Architecture, Yeungnam University, 280, Daehak-Ro, Gyeongsan, Gyeongbuk 38541, Republic of Korea*

Abstract

In this work, an effective Damage-Informed Neural Network (DINN) is first developed to pinpoint the position and extent of structural damage. Instead of resolving the damage identification problem by conventional numerical methods, a Deep Neural Network (DNN) is employed to minimize the loss function which is designed by combining multiple damage location assurance criterion and flexibility matrices to guide the training process. In our computational framework, the parameters of the network, which include both weights and biases, are treated as new design variables instead of damage ratios. Therein, the training data consists only of a set of spatial coordinates of elements, whilst corresponding the damaged ratios of elements unknown to the network are factored into the output. To achieve the goal, the loss value is calculated relying on the predicted damage ratios with supporting Finite Element Analysis (FEA). Additionally, Bayesian Optimization (BO) algorithm is used to automatically tune hyperparameters of the network for enhancing reliability in damage identification. Several numerical examples for damage localization of truss and frame structures are investigated to evaluate the effectiveness and reliability of the suggested methodology. The obtained results point out that our model not only correctly locates the actual damage sites but also requires the least number of structural analyses and faster convergence rate compared with other algorithms.

Keywords: Damage-informed, Damage detection, Hyperparamter tuning, Bayesian optimization, Deep neural network

*Corresponding author. *E-mail: jhlee@sejong.ac.kr*
[1]*E-mail: maitienhaunx@gmail.com, maitienhau@iuh.edu.vn*
[2]*E-mail: seunghye@sejong.ac.kr*

1. Introduction

Over time, several local damages may occur in engineering structures throughout their operational life due to various unforeseen reasons, such as environmental effects, human activities, natural aging process of the materials, and so on. Clearly, the buildup of damage leads to the change in physical attributes of structures and then failures may occur if it is not identified promptly. Therefore, the implementation of structural damage diagnosis is necessary to ensure the safety and usability of structures. And detecting damage at an early stage not only reduces maintenance expenses but also extent the service life.

The important thing that must be highlighted here is that the modal parameters of the structure will change due to the presence of damage [1, 2]. Thus, vibration-based methods have been widely employed for structural damage detection and health monitoring. Among the vibration parameters, mode shape and natural frequency have received much attention from many researchers. Firstly, natural frequency-based algorithms have been successfully applied to detect and localize damage by many researchers [3–6]. As pointed out by previous studies, an essential advantage of this procedure is relatively simple to measure the frequency values with a high level of accuracy [7, 8]. However, the usefulness of these approaches for large structures is restricted due to their low sensitivity to detect local damage. Furthermore, changes of measurement temperatures or structural mass can introduce uncertainties in the measured frequency variations and may lead to undetected damage due to noises and processing errors [9]. To overcome the above challenge, the second one was developed for the identification of damage by analyzing mode shapes or mode shape curvatures obtained through measurements [10–13]. And when it comes to detecting structural damage, mode shapes can give more informative and sensitive than natural frequencies, which makes them a preferred choice [14]. Nevertheless, many sensors located on the structure are required to gain the accuracy. Besides, the damage results can be highly affected by noises and are not significantly influence to the case of low-level damage [8]. Next, one way to combine mode shapes and natural frequencies for damage identification based on the changes in the flexibility matrix was first proposed by Pandey and Biswas [15]. And it demonstrated higher sensitivity to structural defects compared to both above mentioned approaches. Several variations and extensions based on modal

flexibility have been successfully applied for the assessment of structural damage [16–18].

In recent years, a promising alternative is the Finite Element (FE) model updating. This approach utilized optimization algorithms to determine the location and extent of damage by formulating it as an optimization problem [19]. Therein, the objective function is designed to reflect deviations in dynamic characteristics between the FE model and the structure bearing damage [20]. The parameters of the FE model are then locally adjusted to obtain a good agreement with those of the damaged structure by minimizing the objective function. In the context of damage detection, there have been various algorithms proposed to tackle the optimization problems in FE model updating, which can generally be classified into two main groups. The first one includes sensitivity-based methods, which have been successfully introduced for damage detection in a range of structures, such as the coupled local minimizers [21, 22], the trust region Newton [23], the response surface [24], the combining penalty function and random search algorithm [25], the affine scaling interior [26], and so on. However, they exhibit several drawbacks, such as the requirement of calculating the sensitivity matrix, increased computational effort due to large-scale structures, and the possibility of getting stuck in a local optimum related to the highly nonlinear and multimodal problems, as well as depending on the choice of initial point, which can reduce efficiency and even lead to failure [20]. The other group is metaheuristic algorithms, which get their inspiration from natural phenomena such as evolution, swarm intelligence, and population genetics to find the optimal solution without the need for gradient information. There are many algorithms available for structural damage tracking, such as Differential Evolution (DE) [27], Genetic Algorithm (GA) [28], Jaya [14], Particle Swarm Optimization (PSO) [29], Cuckoo Search (CS) [30], Artificial Bee Colony (ABC) [31], teaching learning-based optimization [32], adaptive hybrid evolutionary firefly algorithm [33], Crow Search Algorithm (CSA), Colliding Bodies Optimization (CBO) [68], and so on. Although some success has been attained by these algorithms, they exhibit certain limitations, such as slow convergence rate, requiring a huge amount of numerical simulations, many tuning parameters, and high computational costs especially when dealing with large-scale problems [34].

Over the past few years, Machine Learning (ML) has proven to be a flexible and power-

ful enough instrument for decision-making in various fields, including language processing, medical diagnoses, industrial automation, voice recognition, etc. The deep neural network, a type of machine learning model, has been widely adopted in computational mechanics, such as structural optimization [35–39], structural stability [40], structural analysis [41, 42], reliability analysis [43], and structural health monitoring. Starting from the 1990s, the neural network (NN) was successfully applied for structural damage detection. For instance, an automatic monitoring approach using the NN was delivered by Wu et al. [44] for detection of structural damage. Meanwhile, Tsou and Shen [45] developed a combining model including three NNs for online damage identification of mass-spring systems. In addition, an efficient online diagnostic model based on the NN was suggested by Teboub and Hajela [46] to detect the damage in composite material beams. In recent times, Mehrjoo et al. [47] applied the NN to assess the severity of damage in joints of truss bridge structures. Tam et al. [48] built DNN based on the data training process to predict damage location severity of truss structures. Besides, Parisi et al. [49] adopted ML classification tools for detecting structural damage in steel truss railway bridges. Meanwhile, Wang et al. [50] developed an automatic damage detection system based on ResNet101 model to locate the damage in real-time. A combined model of YOLO-v4 and object detection algorithm to locate concrete wind-erosion damage was suggested by Cui et al. [51]. And Ho et al. [52] attempted to localize damage in a steel beam structures by using hybrid feedforward NNs and the marine predator algorithm. Also, a large variety of ML applications can be found for detecting the damage locations of various structures. It is worth mentioning that most of the existing studies rely on purely data-driven models. Despite the various studies mentioned above, they all operate based on the same fundamental principle, as shown in Fig. 1. According to this flowchart, it can be seen that it consist of two main blocks: one for generating training data and the other for building the machine learning model. For the data preparation phase, in the first step, several sampling techniques are employed to generate a set of input variables $\mathbf{x}$, and then the corresponding output variables $\mathbf{y}$ can be collected through experiments, numerical simulations, or a combination of two methods. The whole training dataset $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\,;\ i = 1,\ 2,\ ...,\ k\}$ is labeled and k indicates the number of the sampling points. Note that there are various different variables that can be chosen as training dataset, but they

should contain the responses of the structure in various damaged states. Next, the NN with the appropriate architecture was designed to establish mappings between the input $\mathbf{x}$ and output data $\mathbf{y}$ within the specific domain defined by the training dataset $\mathbf{D}$. And the approximation in NN is found based on the idea of supervised learning by training to minimize Mean Square Error (MSE) loss which represents the distinction between the network prediction $\hat{\mathbf{y}}$ and the available true data $\mathbf{y}$.

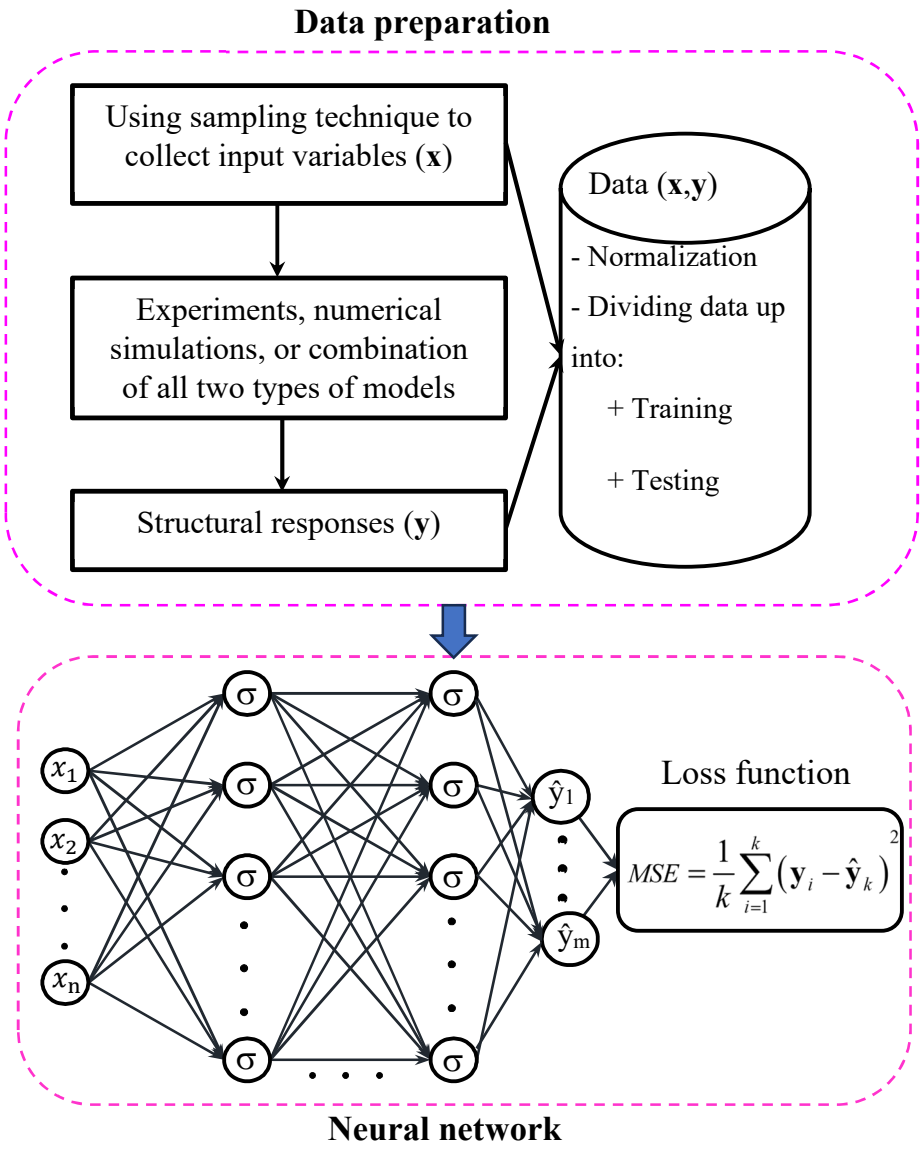


Fig. 1. Flowchart depicting the purely data-driven model for damage identification.

Despite its success in the area of damage identification, the data-driven approach also encounter several challenges as bellow:

(i) Firstly, this framework has used the network as a surrogate model which learned the mul-

tifaceted relationships between the inputs and outputs to predict the structural responses. Therefore, it demands a large number of training data points to produce a reliable model capable of accurately capturing the response[44]. Especially when the complexity of the problem increases, its effectiveness strongly depends on the size and quality of the collected training data [53]. Furthermore, experiments or discretization methods, such as the finite element method, are required to collect the available true data for training the network. Hence, the phase of generating data is usually the bottleneck step in the development of any data-driven model [54]. A challenging question with no easy answer posed here is how to determine the suitable data size for different problems?

(ii) Next, it is easily seen that the norm of deviation or the MSE between the outputs of the network prediction $\hat{\mathbf{y}}$ and the available true data (labeled dataset) $\mathbf{y}$ is derived as the loss function which is minimized by training process to estimate the network parameters. Clearly, the loss function does not include the physics information as well as the damage constitutive equations of the structures. Hence, the network's capability is restricted to reproducing the training dataset exclusively and lacks of generality for various damage detection problems.

(iii) Finally, as pointed out by many researchers [44, 54, 55], the obtained results are very much contingent upon the chosen network architecture as well as the richness of the training data set of the essential information required for damage assessment. Consequently, a key challenge that still remains is how to efficiently find the best hyperparameters of the network without relying on user experience research while still ensuring accurate results.

In contrast to data-driven approaches, a novel neural network model called Physics-Informed Neural Network (PINN) has received considerable attention in the computational mechanics community, in which the network considers the underlying physics of the problem that the governing equations including the initial and boundary conditions are integrated directly into the original loss function. Unlike the purely data-driven models, it do not require any labeled dataset for training [42, 53]. Owing to the above salient advantages of the PINN model, it has been proven to be successful in solving the complex problems, such as structural analysis [56],

fluid mechanics [57], topology optimization [58], and so on. However, a damage-informed NN model detecting the locations and severity of damage in the structure has not yet been studied. Motivated by the foregoing challenges and based on existing works, this study is conducted to address the aforementioned issues.

This article aims to introduce the damage-informed neural network, which is designed to directly detect the location and extent of damage in structures without using any labeled training data obtained by structural analysis. Instead of employing conventional techniques to resolve structural damage concerns, the DNN is designed to identify the damage locations by minimizing the loss function. In this context, the parameters of the network including weights and biases are considered as design variables. The training data only contains coordinate values of elements, while the unknown damage ratios are taken account of output values, which are expressed by the input values and parameters of the network. Accordingly, the structural responses corresponding to the predicted outputs of the network found by FEA are utilized to build the loss function. Besides, Just Another eXtensor (JAX) [59] and Bayesian optimization are applied to automatically calculate sensitivities and tune hyperparameters, respectively. Finally, the damage location and severity of the structure corresponding to the optimal network are immediately indicated at the end of the training process without the need for any algorithms. The applicability, efficiency, and reliability of the present model are demonstrated through five numerical examples. The results indicated by the DINN with different noise levels are compared with conventional algorithms and actual damages.

The remainder of this work is structured as follows. Next, Section 2 describes a robust damage-informed NN framework. In Section 3, several truss and frame structures are tested to demonstrate the accuracy, effectiveness, and robustness of our approach. Finally, Section 4 outlines some of the most important conclusions of the study.

2. Damage-informed neural network framework

In this section, the DINN is developed to directly detect damage location and severity of structures. The overall proposed scheme is illustrated in Fig. 2, in which the parameters $\boldsymbol{\theta}$ are regarded as design variables of the structural damage detection problem, instead of the dam-

age ratios of elements. The input features to the model are the spatial coordinates of elements $\mathbf{X} \in \mathbb{R}^{d\times n_0}$, while the damage ratios are the output values $\mathbf{e} \in \mathbb{R}^{n_0\times 1}$ of the network that are considered as the basic unknowns. The predicted output values $\hat{\mathbf{e}}$ are expressed as a function of $\boldsymbol{\theta}$ and $\mathbf{X}$ by the nonlinear map of the DNN. Based on the predicted damage ratios generated by the DNN, the loss function, which includes the damage location assurance criterion and flexibility matrices, is established based on the structural responses attained corresponding to these predicted values. Finally, the parameters of the network are adjusted to minimize the loss function of the model, which is equivalent to estimating the structural damage. To achieve the optimal DNN model, BO is applied to efficiently determine the best hyperparameters. The following sub-sections represent two main components: building the loss function and automatic tuning of hyperparameters.

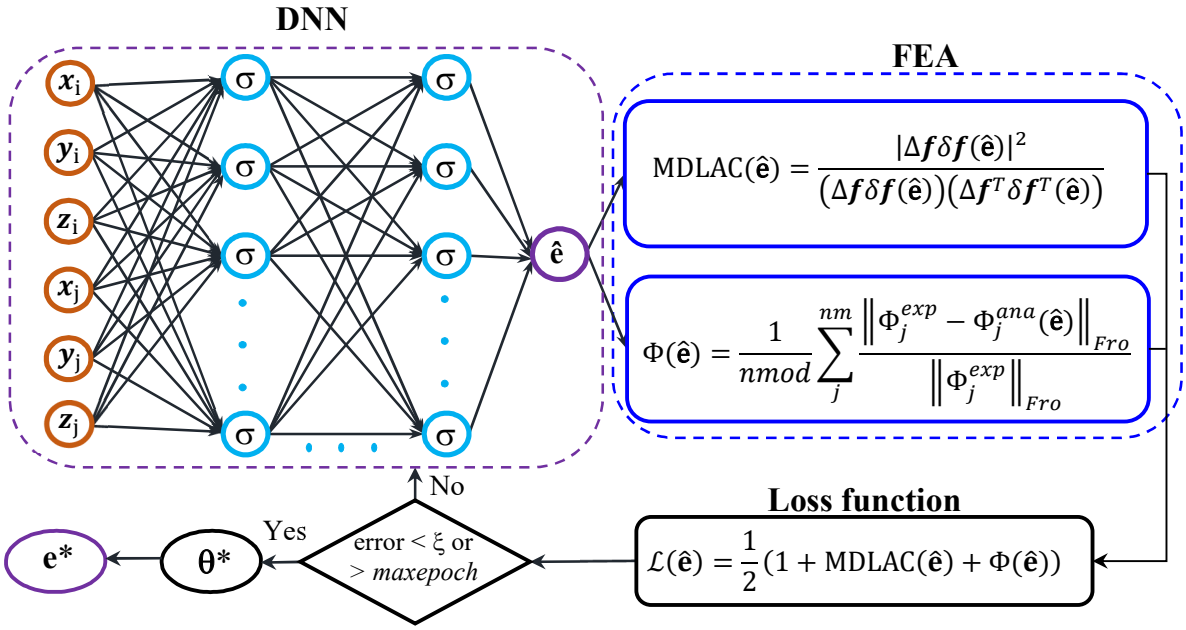


Fig. 2. Flowchart of the DINN framework for damage identification.

2.1. Loss function

Unlike the data-driven approaches, the training data $\mathbf{X} \in \mathbb{R}^{d\times n_0}$ in this study only consists of a collection of spatial coordinates of all elements without the structural responses. Here, n_0 denotes the number of elements, while d equals to 4 or 6, corresponding to two or three dimensions. It is evident that these coordinates can be collected easily from the geometric information of the structure, without the use of any numerical simulations. Meanwhile, the damage ratios,

which are unknown quantities, are not included in the training data. And they are designed as the outputs of the network. It is worth mentioning here that only the predicted values of the network are employed to establish the damage constitutive equations of the structures as the loss function. Clearly, there is a significant difference here between the training data from our approach and the data-driven model. Firstly, it is easily seen that all spatial coordinates of elements are denoted as the whole training data in our model, and obviously it does not require the labeled data and not use any numerical simulations. In other words, our training data does not contain any responses of the undamaged and damaged structures, meaning that they are not completely known prior to the training process. On the contrary for the data-driven models, as shown in Fig. 1, the numerical simulations or experiments are employed to collect the structural responses which are labeled with the known outcomes. Next, the training data is split into two parts for training and testing in the case of the data-driven approach [47], while all spatial coordinates of the structure in the proposed approach is utilized for training and is not devised in itself. Finally, the data size is easily determined from the geometric information of structures for the DINN framework. Meanwhile, this becomes challenging for various problems for the purely data-driven approach [44].

To identify the damage locations, a fully connected DNN with the spatial coordinates of the elements as inputs is built to approximate a solution $\hat{\mathbf{e}}\left(\mathbf{X}, \boldsymbol{\theta}\right)$. Therefore, the estimated damage ratio of the kth element $\hat{e}_k$ can be expressed as follows

$$
\begin{aligned}
&\text{input layer} \quad : \mathbf{h}^0 = \mathbf{X}\left(:,k\right) = \left[x_i^k, y_i^k, z_i^k, x_j^k, y_j^k, z_j^k\right]^T \in \mathbb{R}^6, \\
&\text{hidden layers}: \mathbf{h}^n = \sigma_n\left(\mathbf{W}^n\mathbf{h}^{(n-1)} + \mathbf{b}^n\right) \in \mathbb{R}^{m_n}, \quad for \;\; 1 \leq n \leq (\ell-1), \\
&\text{output layer} \quad : \mathbf{h}^\ell = \hat{e}_k = \sigma_\ell\left(\mathbf{W}^\ell\mathbf{h}^{(\ell-1)} + b^\ell\right) \in \mathbb{R},
\end{aligned}
\tag{1}
$$

where $x_i^k, y_i^k, z_i^k, x_j^k, y_j^k$, and z_j^k are the coordinates at nodes i and j of the kth element; $\mathbf{h}^n$ refers to the output of the layer; $\mathbf{W}$ and $\mathbf{b}$ denote the weight matrix and bias vector, and σ is the activation function that plays a vital in helping the network learn the nonlinear relationship between the input and output. Note that the number of neurons in the input (0th) and output (ℓth) layers are fixed at 6 and 1, respectively. The hyperparameters, including number of hidden

layers ($\ell - 1$), number of neurons in each hidden layer (m), learning rate, activation function, and so on, are found by BO, which are presented in detail in the next subsection.

FEA is utilized to find the mode shapes and natural frequencies with respect to the predicted damage ratios. Based on the achieved structural responses, a hybrid function that combined the natural frequency and modal flexibility change was set up as a loss function of the network [14]. This function was expressed using the parameters and coordinated of the elements, as follows

$$\mathcal{L}\left(\mathbf{X},\boldsymbol{\theta}\right) = \frac{1}{2}\left[1 - \text{MDLAC}\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right) + \text{F}\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right)\right], \tag{2}$$

in which MDLAC(.) and F(.) are the multiple damage location assurance criterion [60] and the modal flexibility-based residual, respectively. They are defined as follows

$$\text{MDLAC}\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right) = \frac{\left|\Delta\mathbf{f}\delta\mathbf{f}\left(\hat{e}\left(\mathbf{X},\boldsymbol{\theta}\right)\right)\right|^2}{\left(\Delta\mathbf{f}^T\Delta\mathbf{f}\right)\left(\delta\mathbf{f}^T\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right)\delta\mathbf{f}\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right)\right)}, \tag{3}$$

$$F\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right) = \frac{1}{n_{\text{ mod}}}\sum_{i}^{nc}\left(\frac{\left\|\mathbf{F}_i^{\text{exp}} - \mathbf{F}_i^{ana}\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right)\right\|}{\left\|\mathbf{F}_i^{\text{exp}}\left(\hat{e}\left(\mathbf{X},\boldsymbol{\theta}\right)\right)\right\|_{Fro}}\right)^2, \tag{4}$$

with

$$\Delta\mathbf{f} = \frac{\mathbf{f}_{ud} - \mathbf{f}_{\text{exp}}}{\mathbf{f}_{ud}}, \quad \delta\mathbf{f}\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right) = \frac{\mathbf{f}_{ud} - \mathbf{f}_{ana}\left(\hat{\mathbf{e}}\left(\mathbf{X},\boldsymbol{\theta}\right)\right)}{\mathbf{f}_{ud}}, \tag{5}$$

where $\Delta\mathbf{f}$ and $\delta\mathbf{f}$ are the frequency change vectors of the measured and analytical result, respectively; $\mathbf{F}^{\text{exp}}$ and $\mathbf{F}^{ana}$ are the flexibility matrices of the damage model and the analytical model; $\|.\|_{Fro}$ denotes the Frobenius norm; n_{mod} and nc represent the number of modes examined and the number of column in the flexibility matrix, respectively; $\mathbf{f}_{ud}$ and $\mathbf{f}_{\text{exp}}$ are the measured frequency for the undamaged and damage structures; and $\mathbf{f}_{ana}$ denotes the analytical frequency vector with respect to the predicted damage extend of element $\hat{\mathbf{e}}$. Interested readers may find more detailed of the derivation in Refs. [14, 60, 61]. Clearly, it can be observed that the damage constitutive equation as shown in Eq. (2) is directly incorporated into the loss function of the network, in which its value is estimated based on the predicted damage ratios

and the corresponding structural responses obtained with supporting FEA. It should be noted that the loss function does not the labeled data. Hence, minimizing the loss is equivalent to satisfying the governing equation which reflects the damage state of the structure. Indeed, this is the major difference between our work and the previous studies.

In order to train the model, the loss function's sensitivity to changes in the parameters is required and it can be easily obtained by using the chain rule to Eq. (2), as follows

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathrm{e}}} \frac{\partial \hat{\mathrm{e}}}{\partial \boldsymbol{\theta}}, \tag{6}$$

in which the first term $\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{e}}}$ is given by

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{e}}} = \frac{1}{2}\left[\frac{-\partial \mathrm{MDLAC}}{\partial \hat{\mathrm{e}}} + \frac{\partial \mathrm{F}}{\partial \hat{\mathrm{e}}}\right]. \tag{7}$$

According to Eq. (6), it can be easily observed that the second term $\frac{\partial \hat{\mathrm{e}}}{\partial \boldsymbol{\theta}}$ is computed automatically by backpropagation algorithm, which is readily integrated into the network. Therein, JAX [59], which is an automatic differentiation tool, is utilized to calculate gradient of the terms $\frac{\partial MDLAC}{\partial \hat{\mathrm{e}}}$ and $\frac{\partial F}{\partial \hat{\mathrm{e}}}$. Therefore, the gradient of the loss function with respect to the parameters of the network is entirely defined. The trainable weights and biases $\boldsymbol{\theta}$ are then adjusted for each iteration of the training process by the optimizer to minimize the loss function. Once the training phase ends, the damage ratios corresponding to the optimal parameters of the network are determined.

2.2. Tuning hyperparameters

Hyperparameter is a group of parameters that deals with the model training and structure of the network. These must be set before the training process and cannot be directly found from the training data. They played a critical role in the effectiveness of utilizing DNNs for real-world applications. However, there is not a closed-form expression for Hyperparameter Optimization (HPO) problem, which is known as an expensive black-box problem to find the extrema. Thus, conventional algorithms are not well-suited for performing such tuning tasks. In particular, this problem can not be solved by gradient-based algorithms because of a lack of gradient information. Another option is the gradient-free algorithms which rely on popula-

tion genetics and evolution. However, these algorithms require a large number of evaluation functions, so it will take more time and more cost-consuming due to the black-box problem. In order to overcome the drawbacks of above optimization algorithms, surrogate model-based optimization methods were developed for solving expensive optimization problems. One of which is Bayesian optimization algorithm based on Gaussian Process (GP) which is known as a powerful and popular tool for optimizing the hyperparameters of machine learning models [62]. Owing to the salient advantages of this algorithm that includes black-box optimization, high-dimensional space, noisy evaluations, global optimum, configurations in complex, and nonlinear, it has been successfully proven the efficiency and robustness for tuning the hyper-parameters of the machine learning models [63]. Consequently, BO is applied to address the automatic tuning problem, which is formulated as follows

$$\boldsymbol{\beta}^{*} = \underset{\boldsymbol{\beta} \in \Omega}{\arg\min} \;\; g\left(\boldsymbol{\beta}\right), \tag{8}$$

where $g\left(\boldsymbol{\beta}\right)$ represents the performance of the network, which corresponds to the minimum values of the loss function for a given set of hyperparameters $\boldsymbol{\beta}$. And it is considered as the objective function for performing the tuning operation.

To achieve this goal, a surrogate probabilistic model based on the GP is utilized to approximate the objective function $g\left(\boldsymbol{\beta}\right)$. In the first step, let us consider p initial observations of the unknown function $g\left(\boldsymbol{\beta}\right)$ as the training set $\{\boldsymbol{\beta}_{1:p},\ g_{1:p}\}$, where the values g_i correspond to the minimum values of the loss function $\mathcal{L}_{min_i}$. The function values $g_{1:p}$ are distributed jointly according to a multivariate Gaussian distribution $g_{1:p} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}\right)$, where $\mathbf{K}$ is the kernel matrix and can be given by

$$\mathbf{K} = \begin{bmatrix} k\left(\boldsymbol{\beta}_1, \boldsymbol{\beta}_1\right) & \cdots & k\left(\boldsymbol{\beta}_1, \boldsymbol{\beta}_n\right) \\ \vdots & \ddots & \vdots \\ k\left(\boldsymbol{\beta}_n, \boldsymbol{\beta}_1\right) & \cdots & k\left(\boldsymbol{\beta}_p, \boldsymbol{\beta}_p\right) \end{bmatrix}, \tag{9}$$

in which k is the covariance function and the Matérn kernel function is selected to compute the covariance value [64]. For a new sample point $\boldsymbol{\beta}_{p+1}$, let g_{p+1} be the minimum loss function value of the network. Both $g_{1:p}$ and g_{p+1} are jointly Gaussian, and they can be written as

$$\begin{pmatrix} g_{1:p} \\ g_{p+1} \end{pmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k\left(\boldsymbol{\beta}_{p+1},\ \boldsymbol{\beta}_{p+1}\right) \end{bmatrix} \right), \tag{10}$$

where $\mathbf{k} = \begin{bmatrix} k\left(\boldsymbol{\beta}_1,\ \boldsymbol{\beta}_{p+1}\right) & k\left(\boldsymbol{\beta}_2,\ \boldsymbol{\beta}_{p+1}\right) & \cdots & k\left(\boldsymbol{\beta}_p,\ \boldsymbol{\beta}_{p+1}\right) \end{bmatrix}$. And the posterior probability distribution at a new hyperparameters $\boldsymbol{\beta}_{p+1}$ can be derived via Bayes' rule as

$$P\left(g_{p+1}|\boldsymbol{\beta}_{p+1},\ \boldsymbol{\beta}_{1:p},\ g_{1:p}\right) \sim \mathcal{N}\left(\mu\left(\boldsymbol{\beta}_{p+1}\right), \sigma^2\left(\boldsymbol{\beta}_{p+1}\right)\right), \tag{11}$$

where the posterior mean $\mu(.)$ and covariance function $\sigma^2(.)$ are given by

$$\begin{aligned} \mu\left(\boldsymbol{\beta}_{p+1}\right) &= \mathbf{k}^T[\mathbf{K} + \sigma^2\mathbf{I}]^{-1} g_{1:p}, \\ \sigma^2\left(\boldsymbol{\beta}_{p+1}\right) &= k\left(\boldsymbol{\beta}_{p+1},\ \boldsymbol{\beta}_{p+1}\right) - \mathbf{k}^T[\mathbf{K} + \sigma^2\mathbf{I}]^{-1}\mathbf{k}. \end{aligned} \tag{12}$$

In order to indicate the next most potential hyperparameters, an acquisition function is built to find the maximum value based on the previous training set $\{\boldsymbol{\beta}_{1:p},\ g_{1:p}\}$. It plays a vital role in the performance of BO, and there are several common acquisition functions, such as the Probability of Improvement (PI), Lower Confidence Bound (LCB), Upper Confidence Bound (UCB), Expected Improvement (EI), entropy search, and so on. Therein, the EI is the most prominent choice for HPO, and has been widely and successfully applied in many studies, thus it is selected for this study. Its mathematical expression is defined as follows

$$EI\left(\boldsymbol{\beta}\right) = \left(\mu\left(\boldsymbol{\beta}\right) - g\left(\boldsymbol{\beta}^+\right)\right)\Phi\left(\xi\right) + \sigma\left(\boldsymbol{\beta}\right)\phi\left(\xi\right), \tag{13}$$

with

$$\xi = \frac{\mu\left(\boldsymbol{\beta}\right) - g\left(\boldsymbol{\beta}^+\right)}{\sigma\left(\boldsymbol{\beta}\right)}, \tag{14}$$

in which $\boldsymbol{\beta}^+ = \underset{\boldsymbol{\beta}_i \in \mathcal{D}}{\arg\min}\ g\left(\boldsymbol{\beta}_i\right)$ is the current best hyperparameters obtained from the approximate function $g\left(\boldsymbol{\beta}_i\right)$; $\phi(.)$ and $\Phi(.)$ denote the probability density function and standard normal cumulative distribution function, respectively.

The overall schematic of the proposed model is illustrated in Fig. 3. The diagram shows two loops. The inner loop is the training process of the network to obtain weights and biases

over the epochs, which provides the minimum loss function value corresponding to the hyperparameters. The minimum value is then passed on to the outer loop, where the hyperparameters of the network are tuned by the BO algorithm. The objective of the second loop is to identify the hyperparameters at which the best minimum loss function value is found. All the main stages of the standard BO framework are summarized in Algorithm 1. Accordingly, BO uses a probabilistic surrogate model based on the GP to model the unknown objective function which is the optimal loss function value. Firstly, a set of initial observations $\mathcal{D} = \{\boldsymbol{\beta}_{1:p}, g_{1:p}\}$ including combinations of hyperparameters from the design space and corresponding to the minimum loss function values are collected by latin hypercube sampling and the training process. Next, the surrogate model is build on $\mathcal{D}$. And then an acquisition function, which is expected improvement Eq. (13), is maximized to select the next hyperparameter configuration $\boldsymbol{\beta}_{n+1}$. When the network with respect the new set of hyperparameters is trained to evaluate the minimum loss function, respectively. And then the new data point is appended to the data $\mathcal{D}$. The surrogate model is updated by retraining with the new data. Steps 7 to 14 are repeated iteratively. At each step, the acquisition function suggests a new set of hyperparameters to evaluate. BO evaluates the objective function, updates the surrogate model. When the stopping criterion is met, the best set of hyperparameters is found corresponding to the smallest value of the loss function.

Algorithm 1 Batch pseudo-code for the standard BO algorithm

Input: p : number of the initial sample point
p_{max}: maximum iterations
Output: $\boldsymbol{\beta}^*, g^*_{min}$: the best solution

1: Using Latin Hypercube Sampling technique to collect n combinations of hyperparameters $\boldsymbol{\beta}_{1:p}$ from the design space
2: Training the network with respect to $\boldsymbol{\beta}_{1:p}$ to find the minimum loss function values $g_{1:p} = \mathcal{L}_{\min_{1:p}}$
3: Collect a set of initial observations $\mathcal{D} = \{\boldsymbol{\beta}_{1:p}, g_{1:p}\}$
4: Current best hyperparamters $\boldsymbol{\beta}^+ = \underset{\boldsymbol{\beta}_i \in \mathcal{D}}{\arg\min}\ g\left(\boldsymbol{\beta}_i\right)$
5: Set $n = p$
6: **while** $n \leq p_{max}$ **do**
7: Find GP's parameters by the maximum likelihood estimation method
8: Construct the GP model on $\mathcal{D}_n$
9: Find $\boldsymbol{\beta}_{n+1}$ by maximizing Eq. (13)
10: Training the network with respect to the hypeparameters $\boldsymbol{\beta}_{n+1}$ to evaluate $g_{n+1} = \mathcal{L}_{\min_{n+1}}$
11: Append $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\boldsymbol{\beta}_{n+1}, g_{n+1})\}$
12: Estimate $\boldsymbol{\beta}^*$
13: Update $\boldsymbol{\beta}^+ = \boldsymbol{\beta}^*$
14: $n = n + 1$
15: **end while**

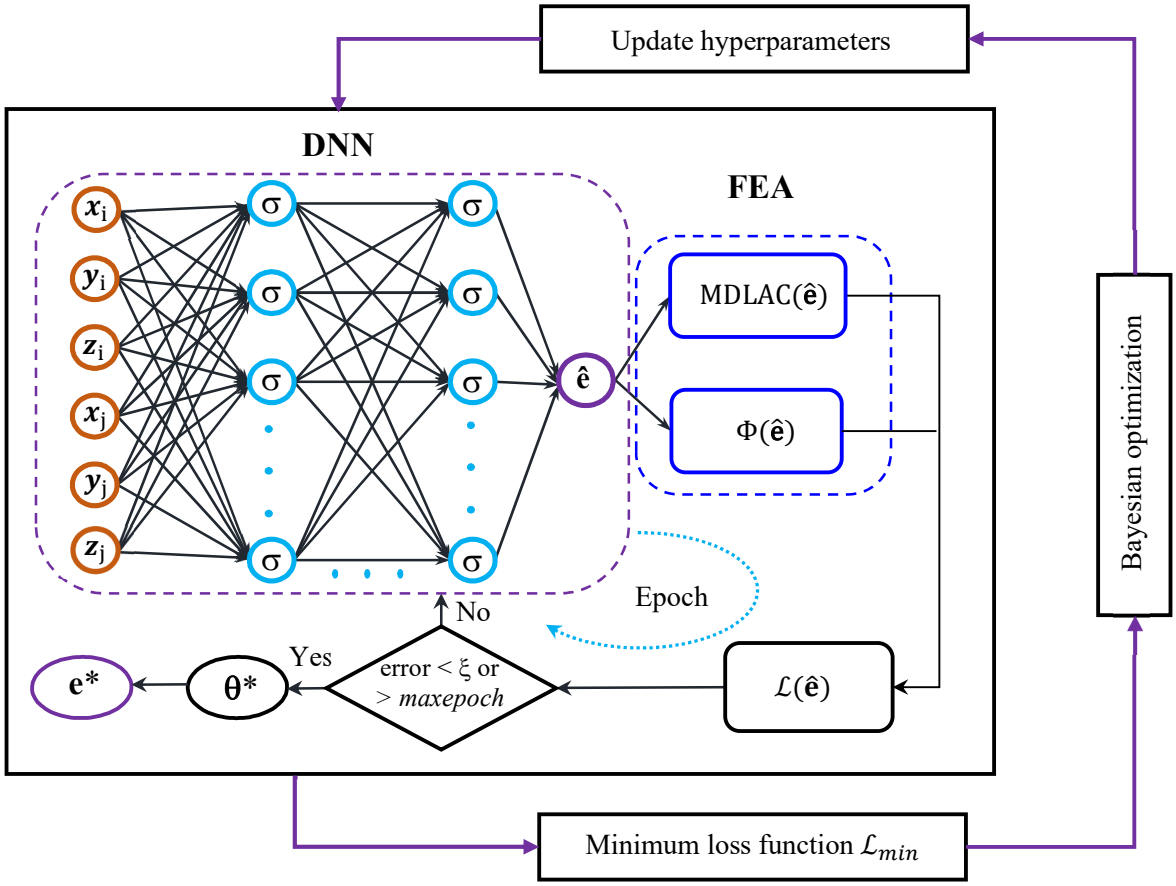


Fig. 3. Schematic of DINN framework into Bayesian optimization for damage identification.

3. Numerical experiments

In this section, several numerical examples are presented to demonstrate the effectiveness of the proposed methodology for damage assessment of truss and frame structures. The obtained damage detection results are compared with those gained from the DE algorithm and previous studies. In order to construct the loss function, free vibration analysis is performed to extract the natural frequencies and mode shapes. To enhance computational efficiency, the BO technique is employed to automatically optimize the hyperparameters of the network. The initial and maximum number of iterations are set to 10 and 30, respectively. Two kinds of hyperparameters are examined in all investigated examples: the first one related to network structure included number of hidden layers, units, and activation function, whilst the remaining one related to training algorithm consisted of parameters to adjust the learning rate. The value range of hyperparameters is listed in Table 1. Adam algorithm and SoftMax are adopted as optimizer and activation function for the output layer, respectively. It should be noted that the learning process stops when either the maximum number of epoch (500) reaches, or the gradient norm of the loss function is less than 0.01 [65]. To assess the impact of uncertain quantities in BO, HPO is implemented in thirty independent runs with different starting points.

For DE algorithm, its control parameters are chosen as follows: population size 50, maximum number of FEAs 3000, stop criterion 10^{-6}, and other parameters are set likewise to Vo-Duy et al [27]. Also, to get the reliable result, the best solution is found by 30 independent runs, due to the stochastic nature of the metaheuristic algorithms. In order to get an impartial comparison between the different algorithms, all of the numerical examples were performed on a personal computer using TensorFlow library with Python language. Additional experiments were conducted using a desktop PC with a 3.0 GHz Intel Core i5-8500 CPU with 16 GB of RAM.

Table 1
Configuration space for the hyperparameters of the network.

Hyperparameter	Search space	Type
No. of hidden layers	[1, 4]	Integer
No. of hidden neurons	[20, 60]	Integer
Activation function	[ReLU, Sigmoid, Softmax, Tanh, LeakyReLU]	Categorical
Learning rate	[0.001, 0.1]	Real
Step size	[1, 10]	Integer
Gama	[0.05, 0.8]	Real

3.1. 21-bar planar truss

A 21-bar planar truss structure, as shown in Fig. 4, is considered as the first numerical example for the structural damage detection. The cross-sectional areas of the members are categorized into 3 groups as following: (1) A_{1-6} = 15 cm^2; (2) A_{7-17} = 9 cm^2; and other members are set as 12 cm^2. The Young's modulus and material density are set as 200 GPa and 7800 kg/m^3 for all truss members, respectively. This structure is investigated for two different damage scenarios, as listed in Table 2. Additionally, to evaluate the influence of noise levels on the accuracy, two noise levels are also considered as following: (1) 0.3% noise in natural frequencies and 5.0% in mode shapes; (2) 0.5% noise in natural frequencies and 8.0% in mode shapes. It should be noted that only the first five mode shapes are utilized for all numerical examples.

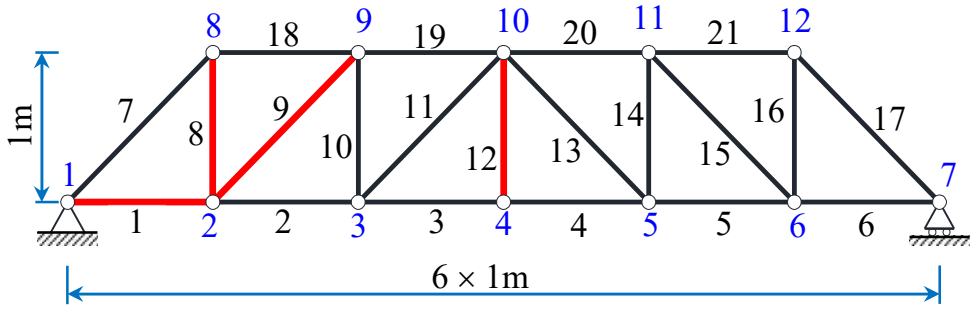


Fig. 4. 21-bar planar truss.

Table 2
Two different damage scenarios in the 21-bar planar truss structure.

Case	1	2		
Element no.	8	1	9	12
Damage ratio	0.25	0.3	0.4	0.3

A comparison of the statistical objective values including best, worst, mean, standard deviation (Std), and 95% Confidence Interval (95% CI) found by the network with using different infill strategies of the BO are tabulated in Table 3. Besides, the optimal hyperparameters of the network are reported in Table 4, respectively. First, it is easily seen that the statistical objective values decreased when the noise level increased. And this result reflected the true nature of the problem. Next, it can be seen that the EI infill strategy achieves the smallest loss function values overall, followed by the PI strategy, and then the LCB strategy. Moreover, the lower bounds of the 95% CI are close to the upper bounds without a significant difference, indicating high reliability of the HPO in identifying the optimal network. Furthermore, the EI infill strategy leads to smaller network architectures with fewer parameters (i.e., weights and biases) compared to the other strategies, as shown in Table 4. The activation functions found by the EI are ReLU and LeakyReLU, which are computationally simple, parameter-free, and prevent gradient saturation, leading to faster convergence than other activation functions. In contrast, the PI and LCB strategies require a larger number of parameters. Finally, Fig. 5 illustrates the best convergence histories of automatic hyperparameters tuning for the scenario with noise level 2 using three different infill strategies. It is evident that the EI strategy achieves faster convergence in detecting the optimal network compared to the other strategies, and hence is used as the infill strategy for the BO in this work.

Table 3
Statistical results of the objective with various infill strategy for both scenarios of the 21-bar planar truss with noise levels.

Infill Strategy	Case	Noise level	Log(objective)				
			Best	Worst	Mean	Std	95% CI
PI	1	0%	-9.1315	-4.2914	-4.6416	-5.1389	[-4.6941, -4.5948]
		5%	-5.4007	-4.2818	-4.5972	-5.2471	[-4.6621, -4.5408]
		8%	-4.8782	-4.0219	-4.3234	-5.0045	[-4.3834, -4.2706]
	2	0%	-6.9997	-5.1319	-5.9331	-6.4500	[-5.9882, -5.8842]
		5%	-5.3469	-4.3726	-4.8235	-5.3587	[-4.9101, -4.7513]
		8%	-5.0168	-4.2727	-4.6003	-5.3535	[-4.6507, -4.5552]
LBC	1	0%	-9.2398	-5.2819	-6.7012	-6.7524	[-6.8674, -6.5813]
		5%	-5.3919	-5.0358	-5.2176	-6.1661	[-5.2491, -5.1883]
		8%	-4.8159	-4.3968	-4.5945	-5.4857	[-4.6305, -4.5612]
	2	0%	-7.7244	-5.6647	-6.8193	-7.0639	[-6.9290, -6.7318]
		5%	-5.4252	-5.2223	-5.3661	-6.6369	[-5.3807, -5.3519]
		8%	-5.1930	-4.5944	-4.9764	-5.7102	[-5.0291, -4.9293]
EI	1	0%	**-14.9848**	-9.2773	-9.9244	-10.4917	[-9.9686, -9.8842]
		5%	**-6.1687**	-5.4083	-5.6735	-6.4154	[-5.7252, -5.6272]
		8%	**-5.6933**	-4.9293	-5.1722	-5.8457	[-5.2334, -5.1186]
	2	0%	**-12.2726**	-7.7644	-8.5539	-8.9663	[-8.6253, -8.4926]
		5%	**-5.6959**	-5.4413	-5.5522	-6.7360	[-5.5701, -5.5349]
		8%	**-5.5247**	-5.2345	-5.3604	-6.4788	[-5.3814, -5.3404]

Table 4
Optimum hyperparameters of the network obtained using the BO with various infill strategy for the 21-bar planar truss.

Case	Noise level	Infill strategy	Hyperparameters					
			Hidden layers	Hidden neurons	Activation function	Learning rate	Step size	Gamma
1	0%	PI	4	52	ReLU	0.0624	8	0.4757
		LCB	4	60	Sigmoid	0.1000	10	0.8000
		EI	3	40	ReLU	0.0960	9	0.2235
	5%	PI	2	44	Softmax	0.0409	8	0.6617
		LCB	2	50	ReLU	0.0324	8	0.5825
		EI	3	25	ReLU	0.0855	5	0.8000
	8%	PI	3	35	ReLU	0.0109	5	0.6004
		LCB	3	35	Softmax	0.0928	8	0.0500
		EI	4	23	ReLU	0.0558	8	0.1107
2	0%	PI	3	46	ReLU	0.0522	3	0.4507
		LCB	4	30	ReLU	0.1000	1	0.0500
		EI	4	20	LeakyReLU	0.1000	3	0.6590
	5%	PI	2	47	Softmax	0.0610	6	0.6250
		LCB	4	30	ReLU	0.0290	3	0.3714
		EI	4	21	ReLU	0.0321	5	0.0500
	8%	PI	2	53	Softmax	0.0301	4	0.1912
		LCB	3	53	ReLU	0.0158	5	0.1392
		EI	4	26	ReLU	0.0619	10	0.5594

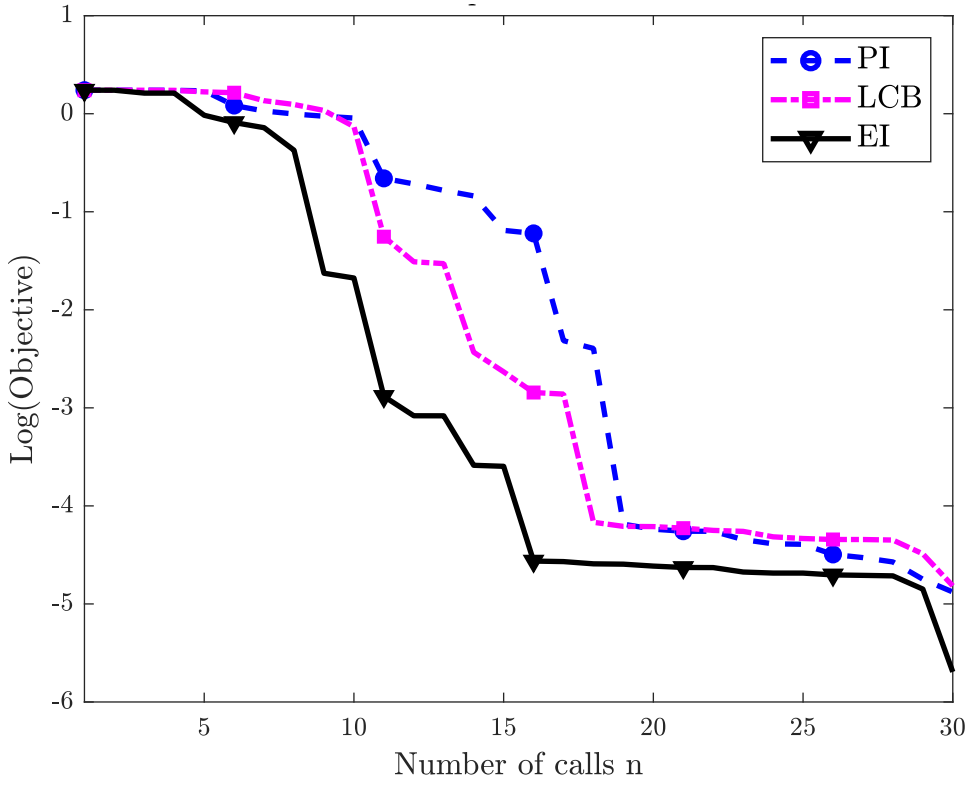


Fig. 5. The convergence histories of the HPO using BO for the scenario 1 of the 21-bar truss with noise level 2.

Table 5 summarized a comparison of the statistical results including the mean, Std, and the average number of structural analysis (Avg) obtained by this work and previous studies for two damage scenarios. For two scenarios without noise, it can be easily seen that the mean damage ratios obtained by the DINN and DE show a quite good agreement with the actual damage ratios, with the errors and Std close to zero, while the CS [14] (0.53%) and Jaya [14] (1.03%) obtained larger errors. And clearly the errors increase with increasing the noise level to 1 and 2, but the DINN performs better than the CS [14], Jaya [14], and DE. In particular, for the noise level 1, the maximum errors found by the DINN, DE, CS [14], and Jaya [14] algorithms are 0.12 %, 6.12 %, 12.56 %, and 13.23 % respectively, whilst those for the noise level 2 are 0.21 %, 19.76 %, 16.16 %, and 14.7 %, respectively. Furthermore, the maximum Std (0.0002) obtained by the proposed model is very small compared with other works. More specifically, concerning the computational cost, the DINN only requires 500 analyses for the performance of all scenarios, while the other algorithms need a large number of FEA calls, and it grows with increasing noise intensity to achieve a good solution. This can be interpreted

by the fact that our paradigm works on the gradient descent method, thereby resulting in a significant reduction in the number of structural analyses. Additionally, it is clear that the mean damage ratios obtained by the present model are quite close to the upper and lower bounds of the 95% CI. Therefore, this demonstrates the efficiency and reliability of the proposed model in structural damage detection.

Figs. 6 and 7 show the loss convergence histories of the DINN and DE for each damage scenarios, respectively. It is observed that the proposed DINN achieves a significantly faster convergence rate compared to the DE algorithm. After only 500 analyses, the loss obtained by the DINN approaches zeros, whereas the DE is still too far behind, requiring 150,000 analyses. Furthermore, the mean damage ratios of all truss members found by the second algorithms for the scenarios 1 and 2 are illustrated in Figs. 8 and 9, respectively. In the noise-free cases as shown in Figs. 8a and 9a, both algorithms could correctly detect all the local damage for both scenarios. However, under conditions of two higher noise levels as shown in Figs. 8b, 8c, 9b, and 9c, the proposed model can exactly determine all damaged members with high precision, while the DE locates the damaged location with the large relative errors compared to the actual damage ratios. Clearly, the obtained results demonstrate the effectiveness and accuracy of the DINN in comparison with conventional algorithms.

Table 5
Statistical results of damage assessment from different optimization algorithms for both scenarios of the 21-bar planar truss with noise levels.

Case	Noise level	Actual Location	CS [14]			Jaya [14]			DE			DINN			
			Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	95% CI
1	0%	a_8	0.2500	0.0000	40,350	0.2505	0.0018	9,035	0.2500	0.0000	56,810	0.2500	0.0000	500	[0.2500, 0.2500]
	5%	a_8	0.2586	0.0433	50,975	0.2586	0.0442	14,145	0.2653	0.0014	150,000	0.2501	0.0001	500	[0.2501, 0.2502]
	8%	a_8	0.2669	0.0973	57,230	0.2678	0.0976	14,410	0.2994	0.0044	150,000	0.2503	0.0002	500	[0.2503, 0.2503]
2	0%	a_1	0.2998	0.0003	29,840	0.2995	0.0002	8,360	0.3000	0.0000	58,276	0.3000	0.0000	500	[0.3000, 0.3000]
		a_9	0.4001	0.0002		0.4000	0.0003		0.4000	0.0000		0.4000	0.0000		[0.4000, 0.4000]
		a_{12}	0.2984	0.0011		0.2969	0.0017		0.3000	0.0000		0.3000	0.0000		[0.3000, 0.3000]
	5%	a_1	0.2927	0.0075	34,010	0.2927	0.0077	10,945	0.3118	0.0013	150,000	0.2996	0.0001	500	[0.2996, 0.2997]
		a_9	0.4060	0.0132		0.4060	0.0128		0.4040	0.0013		0.3997	0.0001		[0.3996, 0.3997]
		a_{12}	0.2623	0.0485		0.2603	0.0513		0.3103	0.0007		0.3000	0.0001		[0.2999, 0.3000]
	8%	a_1	0.2985	0.0188	35,910	0.2984	0.0188	12,135	0.3067	0.0024	150,000	0.2994	0.0001	500	[0.2993, 0.2994]
		a_9	0.4234	0.0203		0.4236	0.0203		0.4052	0.0016		0.3996	0.0001		[0.3996, 0.3997]
		a_{12}	0.2515	0.0808		0.2559	0.0799		0.3176	0.0014		0.2998	0.0001		[0.2997, 0.2999]

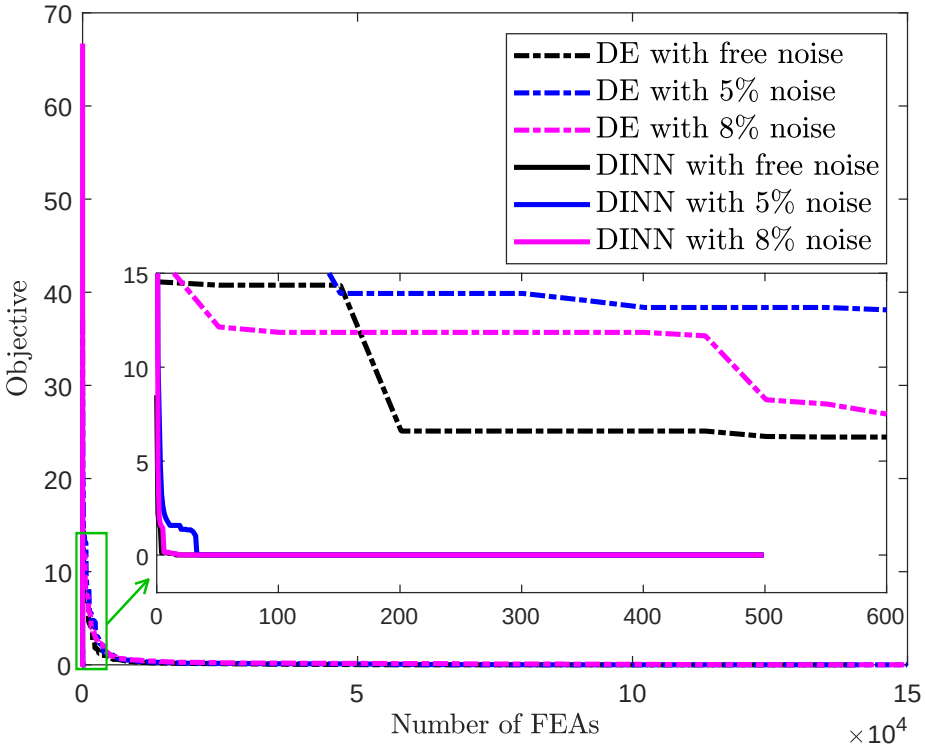


Fig. 6. The convergence histories of the 21-bar truss obtained using the DINN and DE for the scenario 1 with various noise levels.

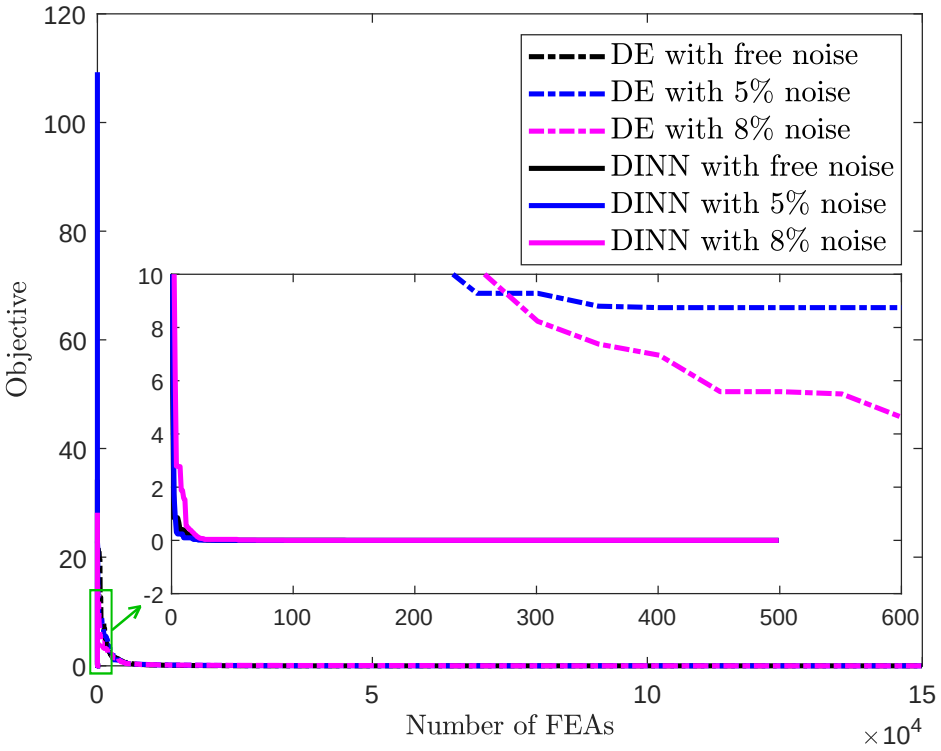


Fig. 7. The convergence histories of the 21-bar truss obtained using the DINN and DE for the scenario 2 with various noise levels.

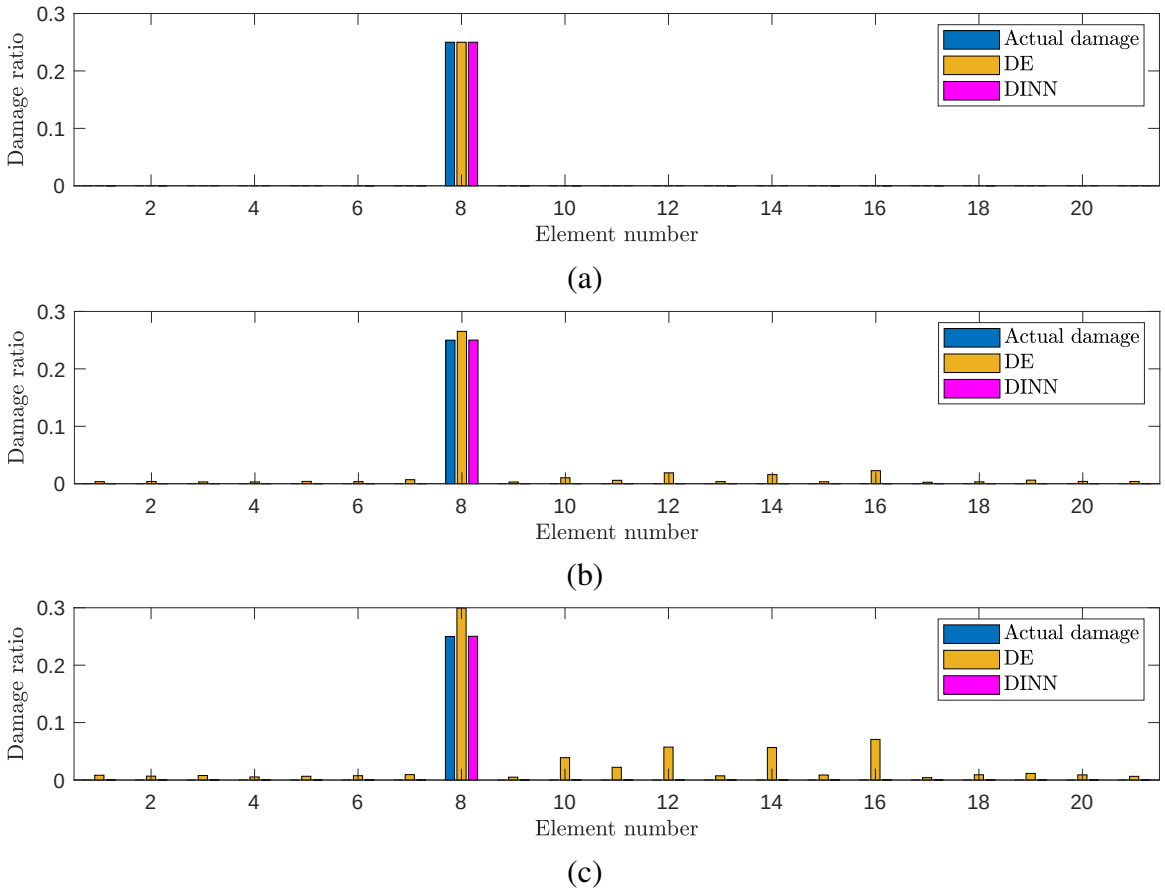


Fig. 8. The damage identification results of the 21-bar truss obtained using the DINN and DE for the scenario 1 with noise levels: (a) noise free; (b) noise level 1; (c) noise level 2.

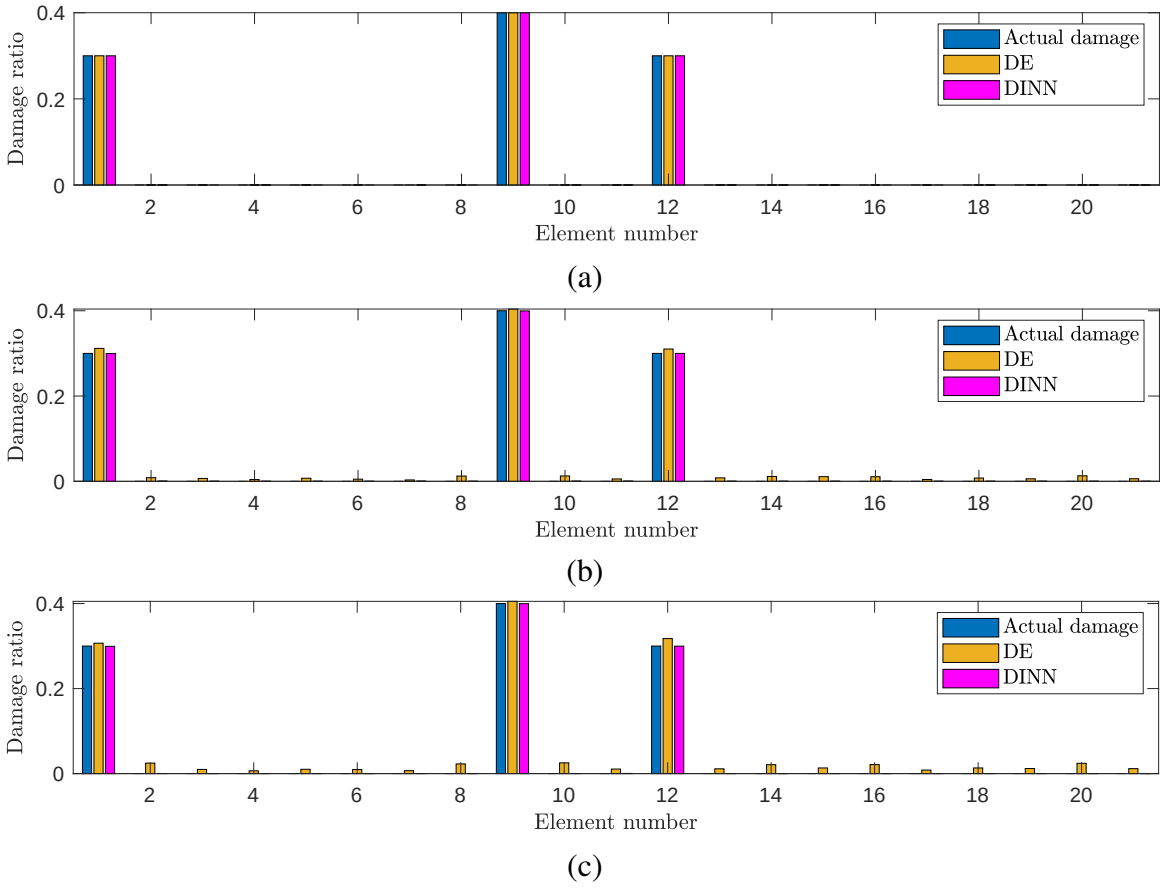


Fig. 9. The damage identification results of the 21-bar truss obtained using the DINN and DE for the scenario 2 with noise levels: (a) noise free; (b) noise level 1; (c) noise level 2.

3.2. 31-bar planar truss

Next, a simple 31-bar planar truss illustrated in Fig. 10 is investigated for the performance of the proposed approach. This problem was first introduced by Messina et al. [60]. The Young's modulus, material density, and cross-sectional area of all truss members are set as E = 2770 kg/m^3, ρ = 70 GPa, and A = 25 cm^2, respectively. In this example, three damage scenarios are considered and the details of damaged elements as well as their corresponding damage ratios for all cases are given in Table 6. In addition, a noise level of 1% in natural frequencies and 5% in mode shapes are added to all damage scenarios.

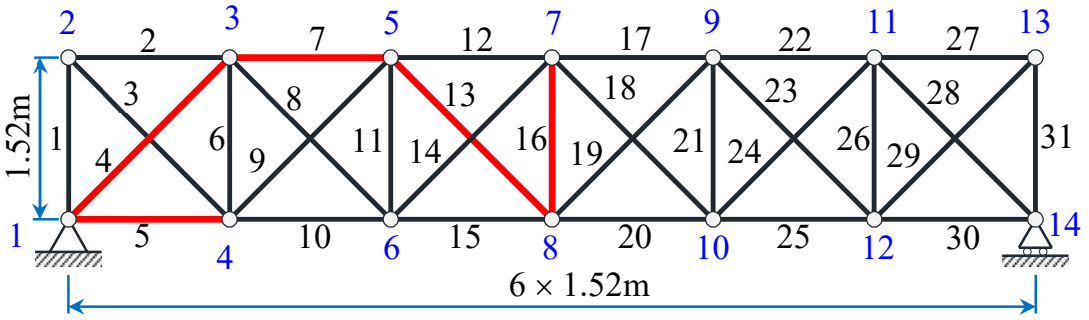


Fig. 10. 31-bar planar truss structure.

Table 6
Three different damage scenarios induced in the 31-bar planar truss structure.

Case	1	2		3		
Element no.	4	5	13	7	13	16
Damage ratio	0.25	0.25	0.35	0.15	0.3	0.4

As the previously presented structure, the optimal hyperparameters of the network in Table 7 are found by the BO with the EI acquisition function for all scenarios. Table 8 and Fig. 11 provide the results achieved by the presented approach and other recently published algorithms in the literature. The data show that the DINN can identify the damaged elements more accurately and precisely than the DE, CS [66], and DSA [66] with the smallest error rate, with the damaged elements located close to the actual damage sites. More concretely, the DE (0.0277) revealed the false alarm 28th member in scenario 1, while the 14th member in scenario 2 were pointed out by the CS (0.0116) and DSA (0.0105). Besides, the standard deviation obtained by the DINN and DE are relatively small compared with the other two algorithms. Furthermore,

there are no significant differences between the average values of damage ratio and the confidence interval limits. Once again, the proposed framework takes only 500 analyses to localize the damage truss members for all scenarios, while the other algorithms require a large number of FEA runs. Finally, a comparison of the convergence curves of both the DINN and DE methods is illustrated in Fig. 12. Clearly, the convergence speed of the learning process is much faster than the conventional DE. Consequently, our model demonstrated again to be the most effective method.

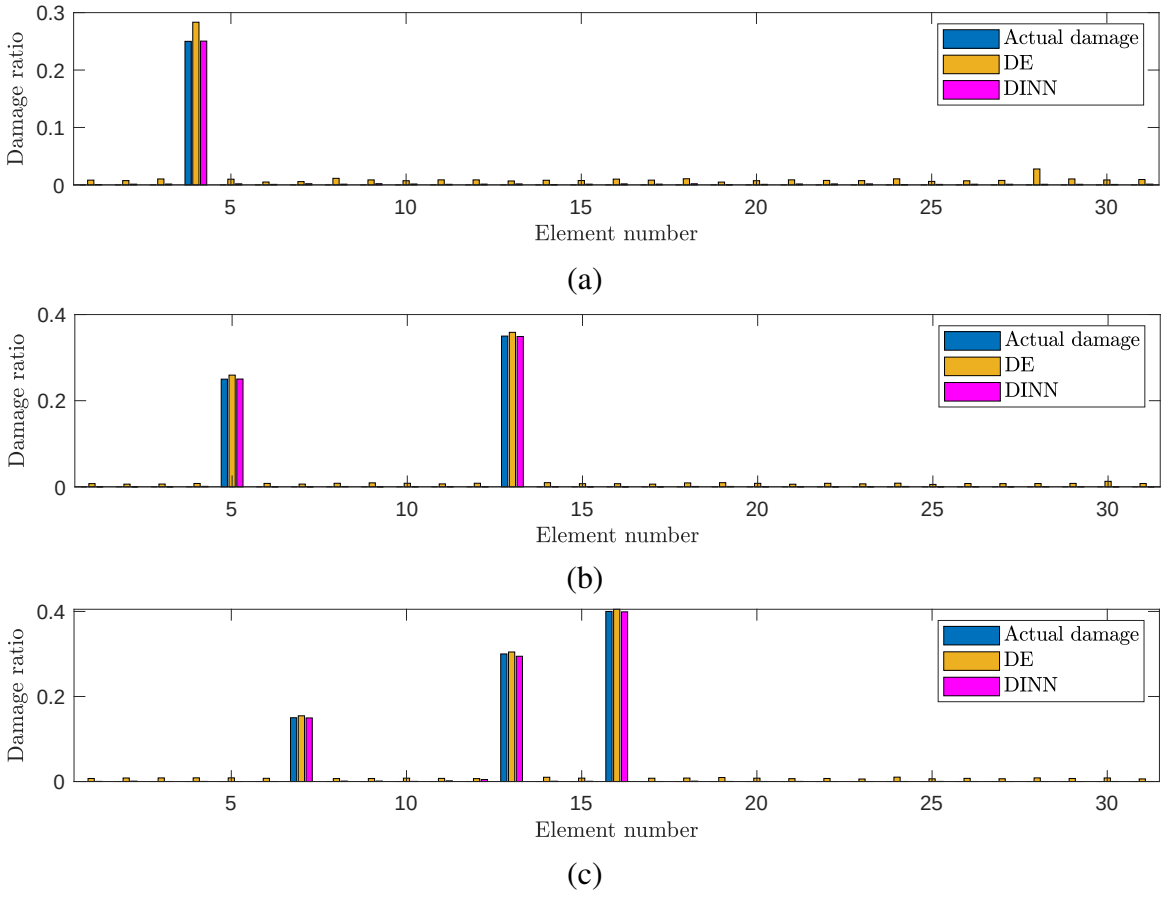


Fig. 11. The damage identification results of the 31-bar truss obtained using the DINN and DE for three scenarios: (a) scenario 1; (b) scenario 2; (c) scenario 3.

Table 7
Optimum hyperparameters of the network obtained using the BO with various damage scenarios for the 31-bar truss structure.

Case	Hyperparameters					
	Hidden layers	Hidden neurons	Activation function	Learning rate	Step size	Gamma
1	3	52	ReLU	0.0088	2	0.7870
2	1	28	ReLU	0.0377	3	0.0500
3	2	23	LeakyReLU	0.0300	6	0.4000

Table 8
Statistical results of damage assessment from different optimization algorithms for three scenarios of the 31-bar planar truss with noises.

Case	Actual Location	CS [66]			DSA [66]			DE			DINN			
		Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	95% CI
1	a_4	0.2513	0.0190	2,868	0.2513	0.0190	392	0.2833	0.0028	150,000	0.2504	0.0002	500	[0.2503, 0.2505]
	a_{28}	-	-	-	-	-	-	0.0277	0.0019		0.0011	0.0007		[0.0006, 0.0014]
2	a_5	0.2661	0.0234	5,520	0.2660	0.0234	1,884	0.2594	0.0014	150,000	0.2503	0.0008	500	[0.2498, 0.2508]
	a_{13}	0.3453	0.0553		0.3460	0.0553		0.3587	0.0012		0.3491	0.0010		[0.3484, 0.3497]
	a_{14}	0.0116	0.0308		0.0105	0.0309		0.0096	0.0005		0.0004	0.0002		[0.0003, 0.0005]
3	a_7	0.2658	0.0350	4,812	0.2658	0.0350	1,788	0.1545	0.0007	150,000	0.1496	0.0019	500	[0.1482, 0.1509]
	a_{13}	0.3759	0.0908		0.3759	0.0908		0.3046	0.0007		0.2946	0.0027		[0.2926, 0.2966]
	a_{16}	0.2029	0.0175		0.2029	0.0175		0.4051	0.0004		0.3990	0.0013		[0.3980, 0.4000]

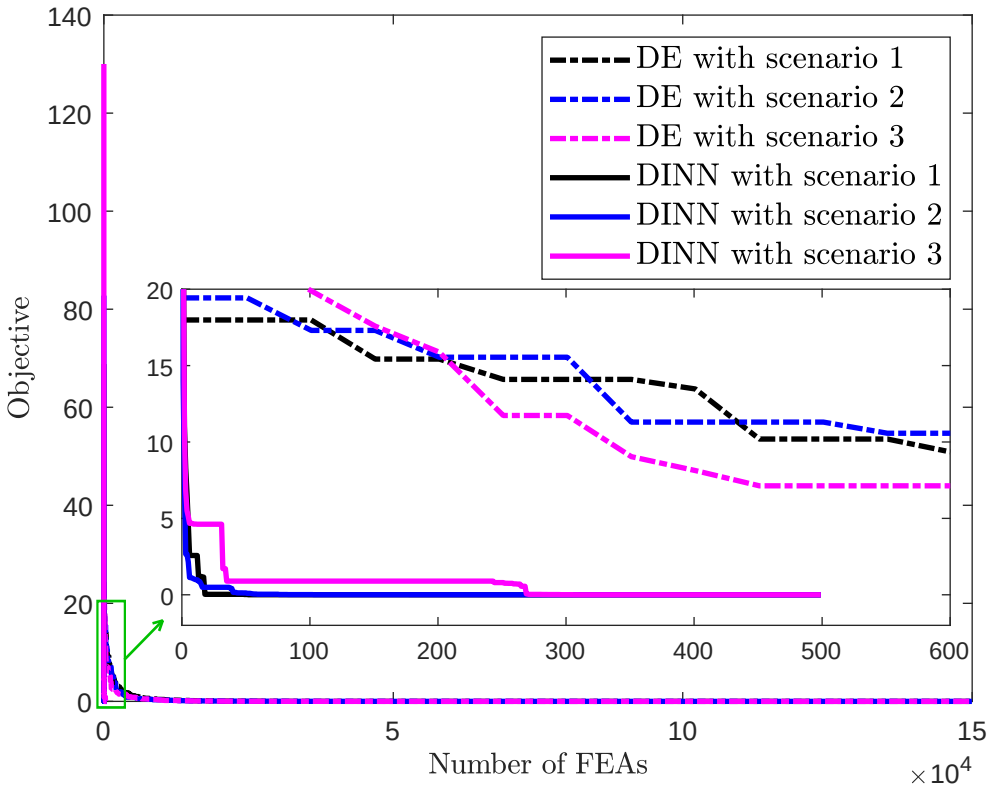


Fig. 12. The convergence histories of the 31-bar truss obtained using the DINN and DE for three scenario with the noise.

3.3. 52-bar dome truss

A 52-bar dome truss is investigated as the third example. The finite element model and coordinates of joints are schematized in Fig. 13. The truss members of the structure are set to the optimal cross-sectional areas obtained by Ho-Huu [67]. The linear elastic modulus and material density are 210 GPa and 7800 kg/m^3 for all truss elements. To evaluate the efficiency of the present methodology, three different damage scenarios are examined with the same noise level as in the second example. These scenarios are listed in Table 9.

Table 9
Three different damage scenarios induced in the 52-bar space truss structure.

Case	1	2		3		
Element no.	9	10	51	9	10	49
Damage ratio	0.25	0.2	0.3	0.2	0.3	0.3

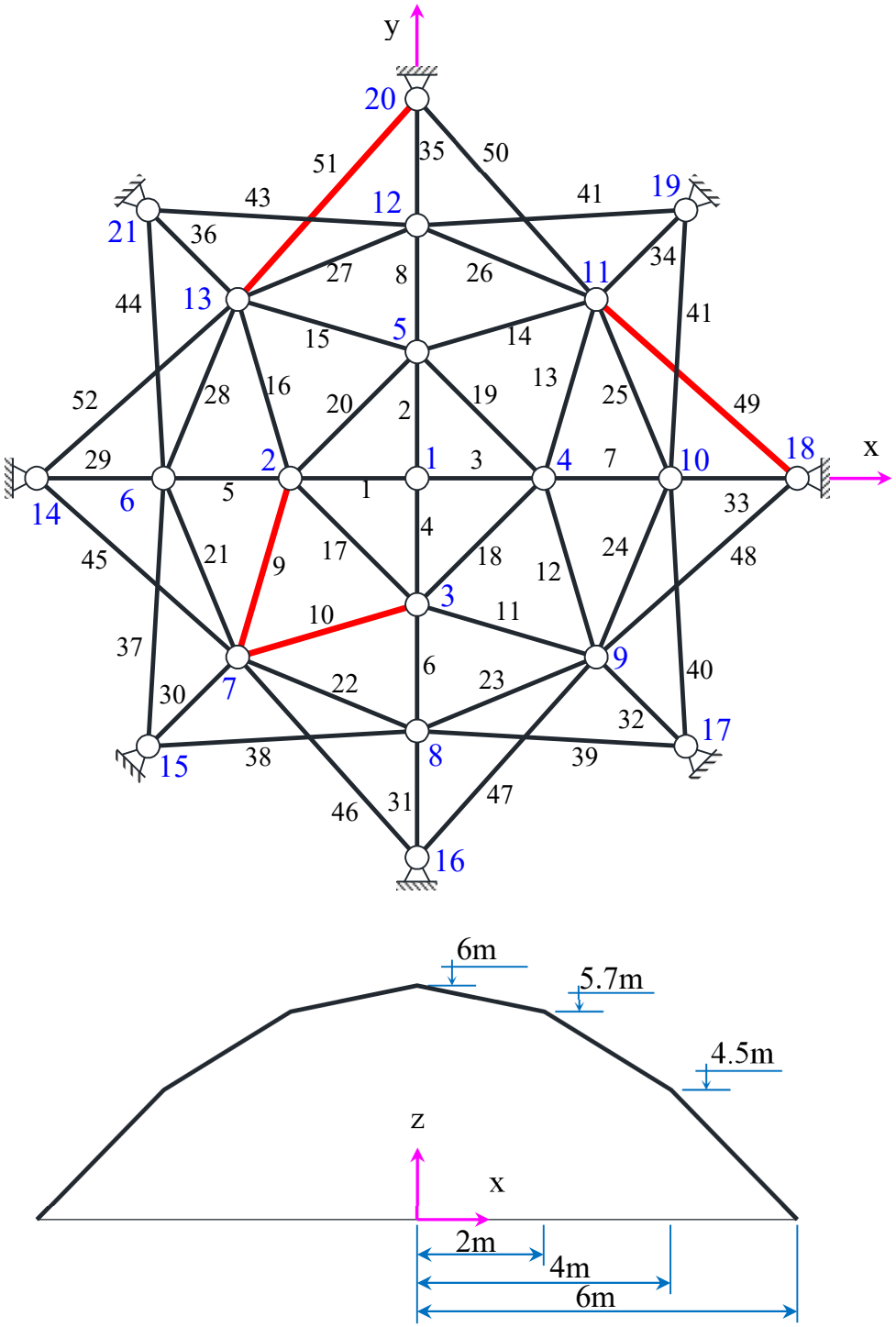


Fig. 13. 52-bar dome truss.

Table 10 presents the optimal hyperparameters of the network obtained for different damage scenarios. A comparison between the identified statistical results of the DINN and other algorithms for the damage assessment is reported in Table 11, with respect to the optimal network. From the data in this table, the results reveal that: 1) the proposed framework can correctly

indicate local damages with the smallest mean error ratios ; 2) the standard deviation gained by the DINN gives the lowest value (except for member 51 in case 2); our approach requires only 500 analyses for all cases, whilst the metaheuristic algorithms need a large number of structural analyses, e.g. CS with 6,834 analyses, ABC with 930 analyses, and DE with 150,000 analyses for the first case; 3) clearly, the confidence limits of the damage ratios found by the DINN are quite close the actual damaged locations with very small deviation. Additionally, Fig. 14 depicts the mean damage ratios of potential damaged members achieved by the DINN as well as the DE algorithm for scenarios 1, 2, and 3, respectively. It is observed that both algorithms give the same precision for determining damage location. Hence, these results again demonstrate the usefulness of the proposed model for structural damage identification.

Table 10
Optimum hyperparameters of the network obtained using the BO with various damage scenarios for the 52-bar space truss.

Case	Hyperparameters					
	Hidden layers	Hidden neurons	Activation function	Learning rate	Step size	Gamma
1	2	54	ReLU	0.0075	2	0.5159
2	3	38	ReLU	0.006	9	0.7245
3	2	54	ReLU	0.0185	1	0.1994

Table 11
Statistical results of damage assessment from different optimization algorithms for three scenarios of the 52-bar space truss with noises.

Case	Actual	CS [66]			ABC [66]			DE			DINN			
	Location	Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	95% CI
1	a_9	0.2475	0.0029	6,834	0.2475	0.0029	930	0.2796	0.0020	150,000	0.2529	0.0012	500	[0.2522, 0.2537]
	a_{20}	0.0000	0.0000		0.0000	0.0000		0.0078	0.0012		0.0024	0.0009		[0.0019, 0.0029]
2	a_{10}	0.2009	0.0018	8,006	0.2009	0.0018	7,682	0.2318	0.0015	150,000	0.2030	0.0013	500	[0.2021, 0.2040]
	a_{14}	0.0033	0.0023		0.0033	0.0023		0.0063	0.0008		0.0006	0.0003		[0.0004, 0.0008]
	a_{43}	0.0324	0.0092		0.0324	0.0092		0.0088	0.0014		0.0012	0.0005		[0.0008, 0.0016]
	a_{51}	0.2788	0.0072		0.2788	0.0072		0.3212	0.0021		0.3020	0.0028		[0.2999, 0.3041]
3	a_9	0.2012	0.0012	5,922	0.2012	0.0012	2,172	0.2172	0.0019	150,000	0.2016	0.0012	500	[0.2007, 0.2024]
	a_{10}	0.3011	0.0015		0.3011	0.0015		0.3151	0.0019		0.3024	0.0006		[0.3020, 0.3028]
	a_{14}	0.0030	0.0025		0.0030	0.0025		0.0119	0.0012		0.0013	0.0003		[0.0011, 0.0015]
	a_{49}	0.3016	0.0026		0.3016	0.0026		0.2874	0.0032		0.2988	0.0015		[0.2979, 0.2998]

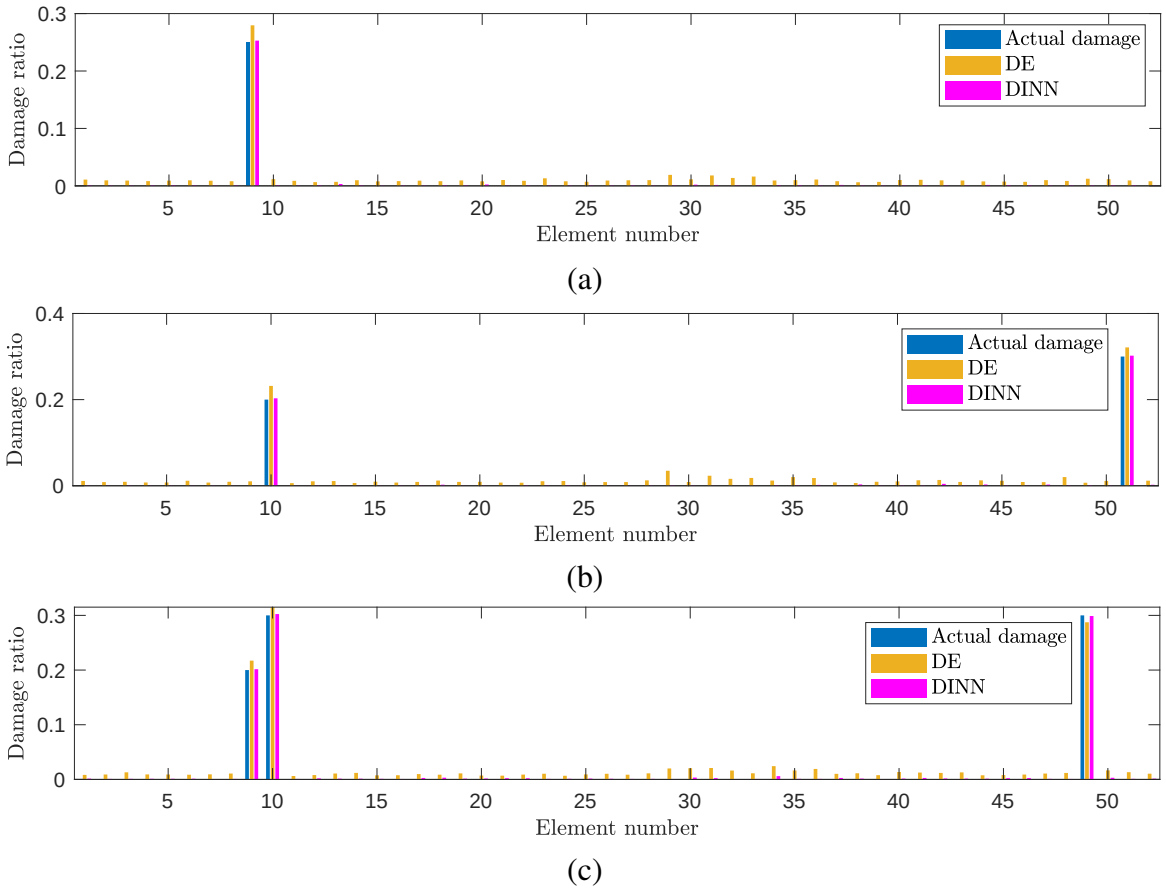


Fig. 14. The damage identification results of the 52-bar truss obtained using the DINN and DE for three scenarios: (a) scenarios 1; (b) scenarios 2; (c) scenarios 3.

3.4. 120-bar dome truss

The next benchmark deals with the damage detection of a large-scale truss structure as shown in Fig. 15. The density density, Young's modulus, and cross-sectional area of all members are set as 7971.81 kg/m^3, 2.1$\times$ 10^{11} N/m^2, and 2.5 cm^2, respectively. A lumped mass of 3000 kg is attached to node one. Nodes 2-13 are connected to a lump mass of 500 kg each, while the remaining nodes have a lump mass of 100 kg each. To assess the effectiveness of the proposed approach, three different damage scenarios free noise, as shown in Table 12, are considered with multiple severity of damage.

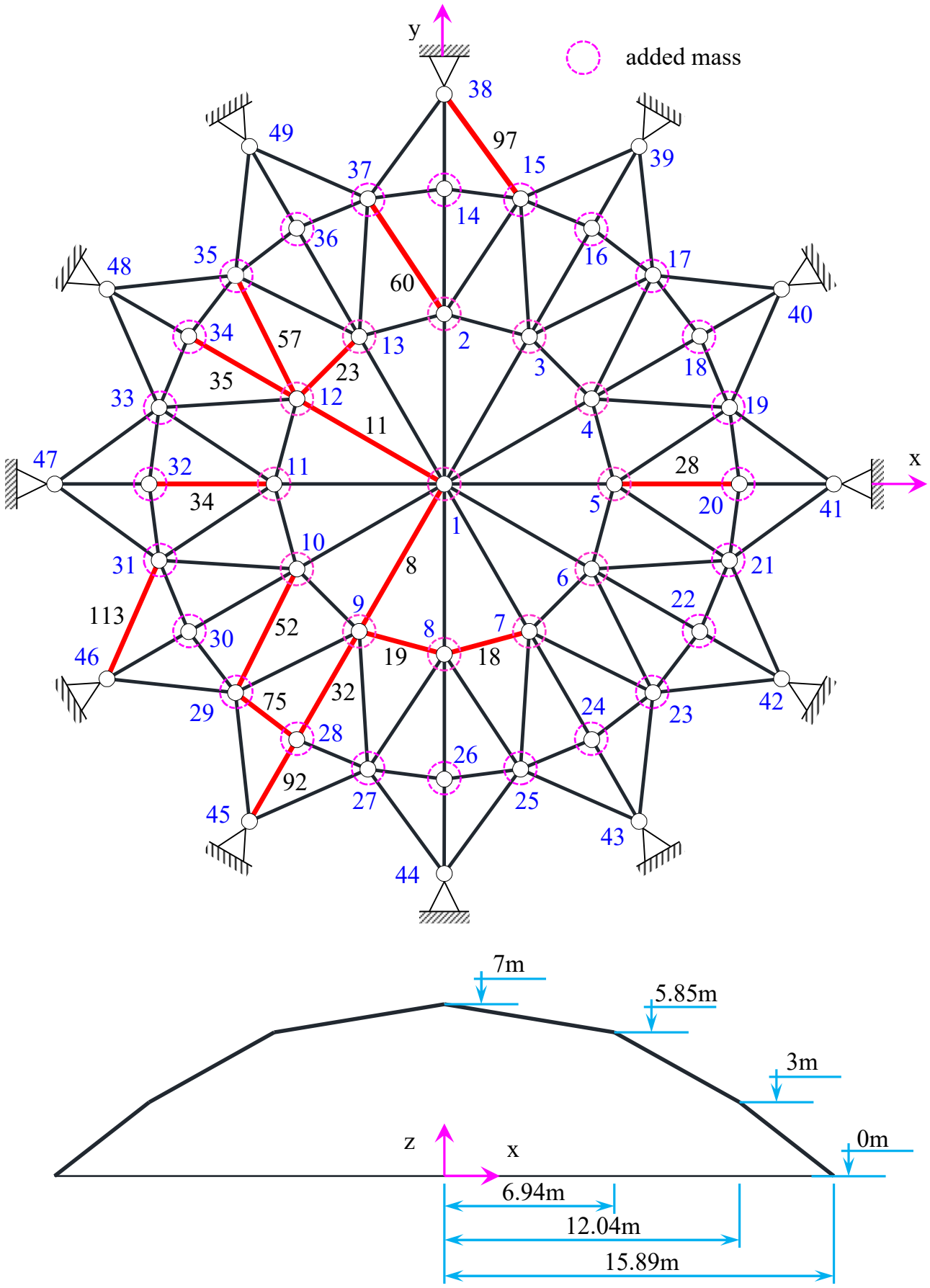


Fig. 15. 120-bar dome truss.

Table 12
Three different damage scenarios induced in the 120-bar dome truss.

Scenario	Element no.	Damage ratio
1	8, 18, 28, 32	0.25, 0.1, 0.2, 0.2
2	11, 19, 23, 32, 57, 75	0.2, 0.05, 0.2, 0.25, 0.15, 0.1
3	11, 34, 35, 52, 60, 92, 97, 113	0.05, 0.1, 0.15, 0.15, 0.1, 0.35, 0.1, 0.15

Table 13
Optimum hyperparameters of the network obtained using the BO with various damage scenarios for the 120-bar dome truss.

Case	Hyperparameters					
	Hidden layers	Hidden neurons	Activation function	Learning rate	Step size	Gama
1	2	54	LeakyReLU	0.0022	4	0.9574
2	3	32	ReLU	0.0829	3	0.0612
3	3	36	ReLU	0.0278	4	0.6824

The optimal hyperparameters results of the network attained by the BO for three damage scenarios is illustrated in Table 13. In addition, Tables 14, 15, 16, and Fig. 16 present the statistical results of damage assessment for three cases. Once more, the obtained damage ratios are detected by DINN with smallest error ratios. More concretely, the maximum error associated with actual damaged elements is approximately 0.03, 0.05, 0.07, 0.129, and 0.002 CSA, PSO, GA, CBO, and DINN, respectively. Furthermore, the $95\%CI$ values found by DINN are quite close the real damage locations with very small deviation. Clearly, it is easily seen that the DINN performs better than the others in terms of the accuracy of damage localization and number of structural analysis. Our model rapidly indicates the damage locations with only 500 FEAs, whilst the others take 20,000. Hence, the proposed approach shows a significant saving of computational efforts when the complexity of problems is increased.

Table 14
Statistical results of damage assessment for first scenario of the 120-bar dome truss.

Algorithm		Elements and damage severities						
		8	18	28	32	40	74	75
CSA [68]	Best	0.2500	0.1010	0.2000	0.2000	0.0000	0.0000	0.0000
	Mean	0.2500	0.1020	0.1830	0.2000	0.0000	0.0000	0.0000
GA [68]	Best	0.2500	0.0970	0.1440	0.2020	0.0000	0.0000	0.0000
	Mean	0.2460	0.1240	0.1300	0.1850	0.0000	0.0360	0.0000
PSO [68]	Best	0.2500	0.1020	0.2000	0.2000	0.0000	0.0000	0.0000
	Mean	0.2500	0.1030	0.1800	0.2000	0.0000	0.0000	0.0000
CBO [68]	Best	0.2500	0.1020	0.1970	0.2010	0.0000	0.0000	0.0000
	Mean	0.2500	0.0990	0.1220	0.1990	0.0000	0.0000	0.0000
DINN	Best	0.2500	0.1000	0.2000	0.2000	0.0000	0.0000	0.0000
	Mean	0.2500	0.0999	0.2000	0.2000	0.0000	0.0000	0.0000
	Std	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
	95% CI	0.2500	0.0999	0.2000	0.2000	0.0000	0.0000	0.0000
		0.2500	0.0999	0.2000	0.2000	0.0000	0.0000	0.0000

Table 15
Statistical results of damage assessment for second scenario of the 120-bar dome truss.

Algorithm		Elements and damage severities								
		11	18	19	23	32	57	73	74	75
CSA [68]	Best	0.200	0.000	0.047	0.203	0.251	0.153	0.000	0.000	0.095
	Mean	0.200	0.018	0.037	0.208	0.250	0.151	0.013	0.016	0.085
GA [68]	Best	0.200	0.033	0.038	0.215	0.250	0.152	0.000	0.060	0.061
	Mean	0.200	0.062	0.029	0.199	0.250	0.151	0.010	0.059	0.089
PSO [68]	Best	0.200	0.000	0.048	0.200	0.252	0.150	0.000	0.000	0.099
	Mean	0.200	0.000	0.049	0.210	0.250	0.154	0.000	0.000	0.100
CBO [68]	Best	0.200	0.000	0.045	0.200	0.250	0.150	0.000	0.000	0.100
	Mean	0.200	0.014	0.033	0.206	0.250	0.150	0.010	0.013	0.088
DINN	Best	0.200	0.000	0.049	0.199	0.250	0.149	0.000	0.000	0.098
	Mean	0.201	0.000	0.050	0.200	0.252	0.149	0.000	0.001	0.099
	Std	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.001
	95% CI	0.201	0.000	0.050	0.200	0.251	0.149	0.000	0.001	0.098
		0.201	0.000	0.050	0.200	0.252	0.150	0.001	0.001	0.099

Table 16
Statistical results of damage assessment for third scenario of the 120-bar dome truss.

Algorithm		Elements and damage severities											
		11	17	18	19	34	35	51	52	60	92	97	113
CSA [68]	Best	0.046	0.010	0.000	0.000	0.109	0.148	0.000	0.150	0.091	0.341	0.103	0.142
	Mean	0.041	0.023	0.025	0.022	0.112	0.150	0.019	0.147	0.100	0.323	0.096	0.123
GA [68]	Best	0.056	0.013	0.042	0.047	0.083	0.152	0.000	0.154	0.114	0.360	0.082	0.161
	Mean	0.052	0.069	0.023	0.098	0.151	0.151	0.067	0.120	0.115	0.303	0.076	0.073
PSO [68]	Best	0.051	0.000	0.000	0.000	0.094	0.148	0.011	0.148	0.100	0.351	0.101	0.149
	Mean	0.049	0.000	0.000	0.000	0.118	0.149	0.030	0.146	0.100	0.300	0.101	0.138
CBO [68]	Best	0.049	0.000	0.010	0.000	0.176	0.150	0.014	0.146	0.100	0.311	0.102	0.132
	Mean	0.050	0.019	0.000	0.025	0.229	0.150	0.024	0.142	0.102	0.272	0.097	0.120
DINN	Best	0.050	0.000	0.000	0.000	0.100	0.150	0.000	0.150	0.099	0.350	0.099	0.150
	Mean	0.050	0.000	0.000	0.000	0.099	0.150	0.001	0.150	0.100	0.351	0.100	0.151
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	95% CI	0.049	0.000	0.000	0.000	0.099	0.150	0.000	0.150	0.100	0.351	0.100	0.151
		0.050	0.000	0.000	0.000	0.099	0.150	0.001	0.150	0.100	0.351	0.100	0.151

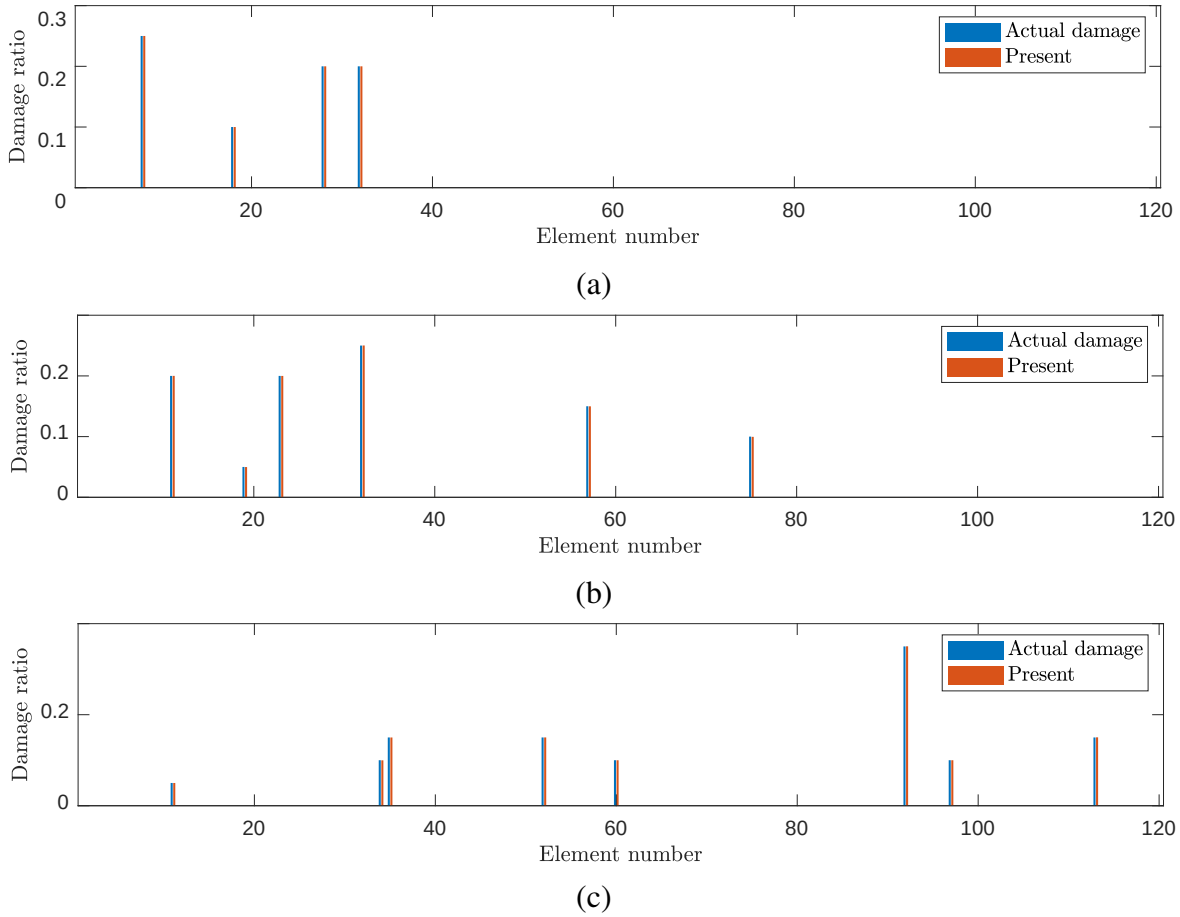


Fig. 16. The damage identification results of the 120-bar dome truss obtained using the DINN: (a) scenario 1; (b) scenario 2; (c) scenario 3.

3.5. A 24-element 2D frame

A 24-element 2D frame structure is considered in the next numerical example. The element representation, boundary condition, and geometry are schematized in Fig. 17. The cross-sectional areas of beam and columns are set to A = 9 cm^2. The material density and Young's modulus of the structure are ρ = 2210 kg/m^3 and E = 71 GPa, respectively. Two hypothetical damage scenarios are examined herein, and the details of damaged members for each case are listed in Table 17.

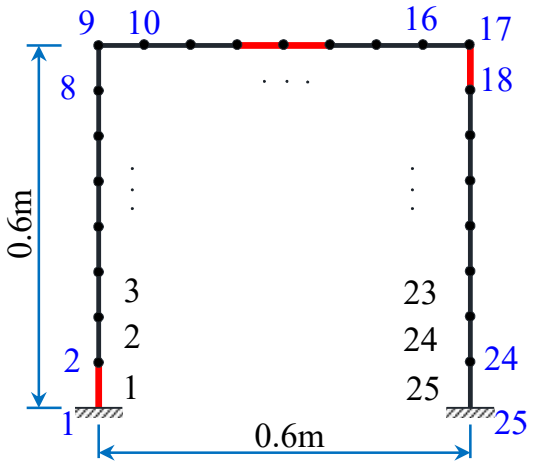


Fig. 17. A 24-element 2D frame.

Table 17
Three different damage scenarios induced in the 24-element 2D frame.

Case	1	2		
Element no.	13	1	12	17
Damage ratio	0.2	0.2	0.3	0.2

Table 18
Optimum hyperparameters of the network obtained using the BO with various damage scenarios for the 24-element 2D frame.

Case	Noise level	Hyperparameters					
		Hidden layers	Hidden neurons	Activation function	Learning rate	Step size	Gamma
1	0%	4	53	ReLU	0.0854	2	0.7959
	5%	3	40	ReLU	0.0564	3	0.3553
	8%	4	30	ReLU	0.0200	3	0.0950
2	0%	4	30	ReLU	0.0200	3	0.2946
	5%	4	23	LeakyReLU	0.0084	8	0.6622
	8%	4	20	ReLU	0.0179	1	0.8000

Similar to the previously investigated example, Table 18 presents the optimal hyperparameters of the network for both cases with different noise levels, as found by the BO using the EI infill strategy. And Figs. 18 and 19 depict the mean values of damage ratio found by two different algorithms for each scenario, respectively. It can be observed that both algorithms can precisely indicate all damaged members for all scenarios, even in the presence of relatively high noise levels. However, our model achieves higher accuracy than the DE for damage identification when the noise level increases. To give a more detailed performance, Table 19

provides a comparison of the statistical results between the DINN and other studies available in the literature. Again, it can be seen that DINN can accurately pinpoint the locations of damaged bars with very small error ratios for all scenarios. Furthermore, the standard deviation results from our approach exhibit the smallest values (except for the scenarios without noises). And it is evident that the application of the network for the damage assessment leads to a reduction of computational cost. Specifically, this work requires only 500 analyses for the convergence performance, while the metaheuristic algorithms require more than ten times as many FEAs to convergence. Based on these results, our method has proven to be not only effective and robust for the damage identification of truss structures but also capable of yielding high accuracy with frame structures.

Table 19
Statistical results of damage assessment from different optimization algorithms for three scenarios of the 24-element 2D frame with various noise levels.

Case	Noise level	Actual Location	CS [14]			Jaya [14]			DE			DINN			
			Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	95% CI
1	0%	a_1	0.2000	0.0000	30,780	0.1999	0.0002	5,765	0.2000	0.0000	77,438	0.1996	0.0002	500	[0.1994, 0.1997]
	5%	a_1	0.1959	0.0085	42,635	0.1959	0.0085	8,020	0.2029	0.0004	150,000	0.1998	0.0002	500	[0.1997, 0.2000]
	8%	a_1	0.1831	0.0262	47,210	0.1829	0.0264	8,795	0.2052	0.0007	150,000	0.2006	0.0003	500	[0.2004, 0.2008]
2	0%	a_1	0.1999	0.0002	31,765	0.1994	0.0003	7,850	0.2000	0.0000	119,336	0.2001	0.0002	500	[0.2000, 0.2003]
		a_{12}	0.3000	0.0002		0.3002	0.0004		0.3000	0.0000		0.2997	0.0001		[0.2996, 0.2998]
		a_{17}	0.1998	0.0002		0.1996	0.0007		0.2000	0.0000		0.2003	0.0002		[0.1998, 0.2008]
	5%	a_1	0.1914	0.0099	40,550	0.1916	0.0103	8,405	0.2023	0.0003	150,000	0.2001	0.0002	500	[0.2000, 0.2003]
		a_{12}	0.3096	0.0203		0.3097	0.0217		0.3018	0.0003		0.2997	0.0002		[0.2996, 0.2998]
		a_{17}	0.1966	0.0360		0.1972	0.0344		0.2023	0.0003		0.1999	0.0001		[0.1998, 0.2000]
	8%	a_1	0.1812	0.0204	41,630	0.1811	0.0205	8,910	0.2032	0.0005	150,000	0.1999	0.0003	500	[0.1997, 0.2001]
		a_{12}	0.2988	0.0203		0.2978	0.0207		0.3025	0.0005		0.2992	0.0004		[0.2989, 0.2995]
		a_{17}	0.1819	0.0170		0.1827	0.0168		0.2037	0.0005		0.2002	0.0003		[0.2000, 0.2004]

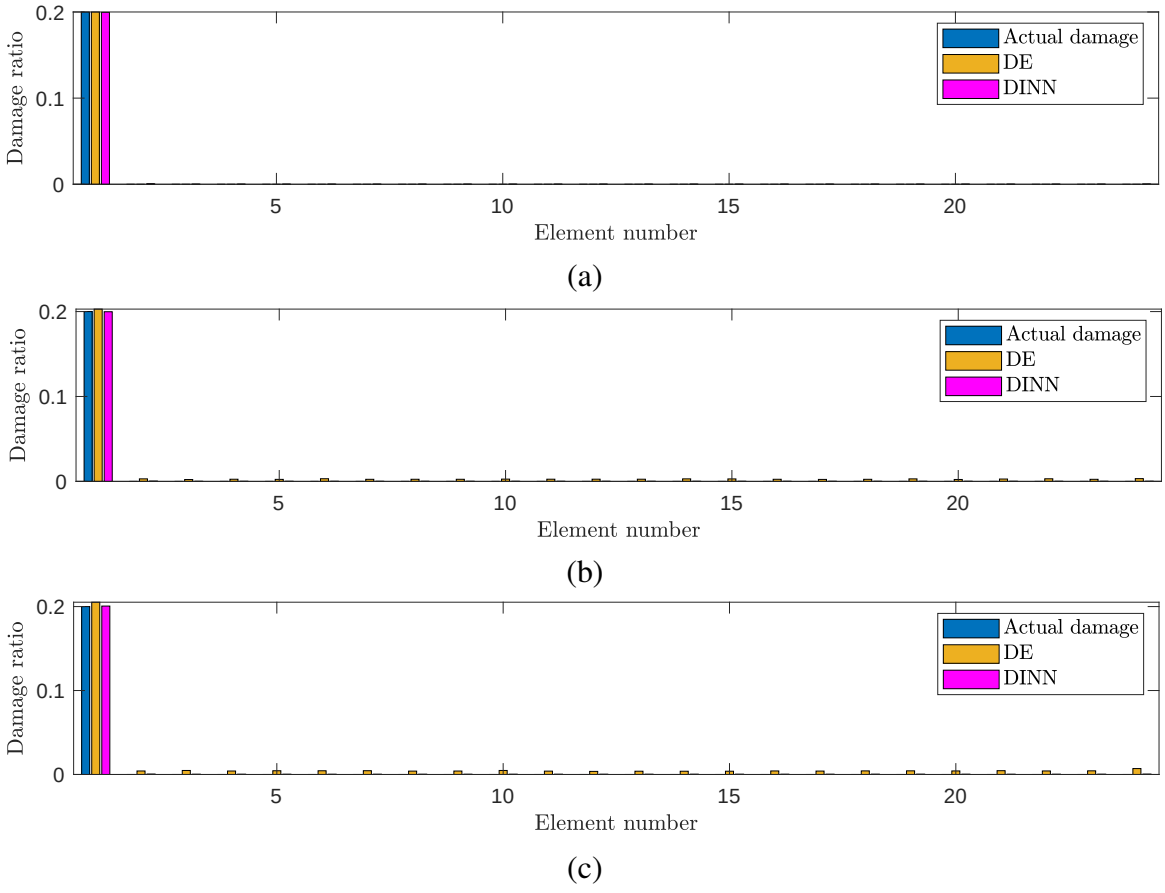


Fig. 18. The damage identification results of the 24-element 2D frame obtained using the DINN and DE for the scenario 1: (a) noise free; (b) noise level 1; (c) noise level 2.

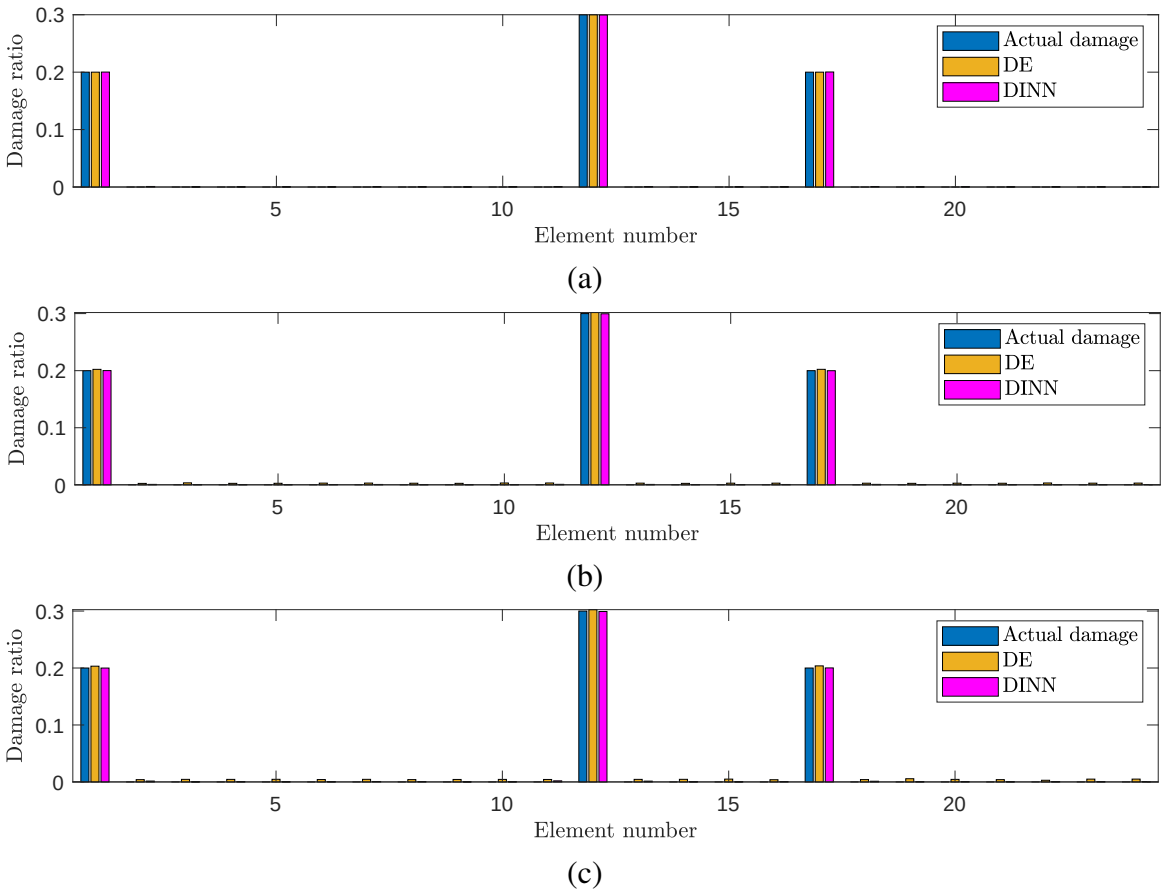


Fig. 19. The damage identification results of the 24-element 2D frame obtained using the DINN and BSA for the scenario 2: (a) noise free; (b) noise level 1; (c) noise level 2.

3.6. A four-storey structure

To demonstrate the accuracy and efficiency of the DINN, a four-storey frame structure was considered as the last numerical example. The geometry, finite element representation, and boundary conditions of the structure are shown in Fig. 20. All frame members are set the same the cross-section of A =0,04 m^2 and are made of steel with the Young's modulus E = 200 GPa and mass density $\rho = 7860$ kg/m^3. Two damage scenarios including single and multiple damage elements given in Table 20 are considered for this benchmark.

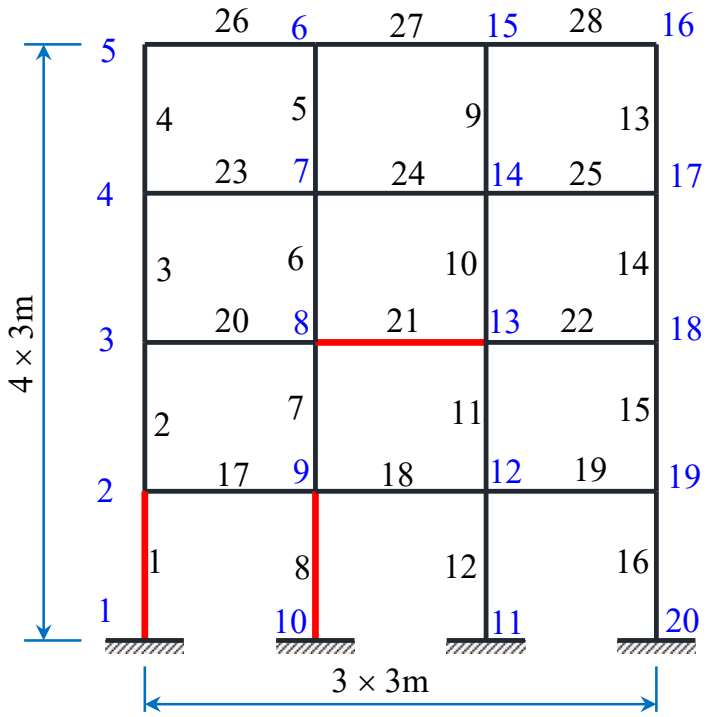


Fig. 20. A four-storey structure.

Table 20
Three different damage scenarios induced in the four-storey structure.

Case	1	2		
Element no.	1	1	8	21
Damage ratio	0.25	0.2	0.3	0.2

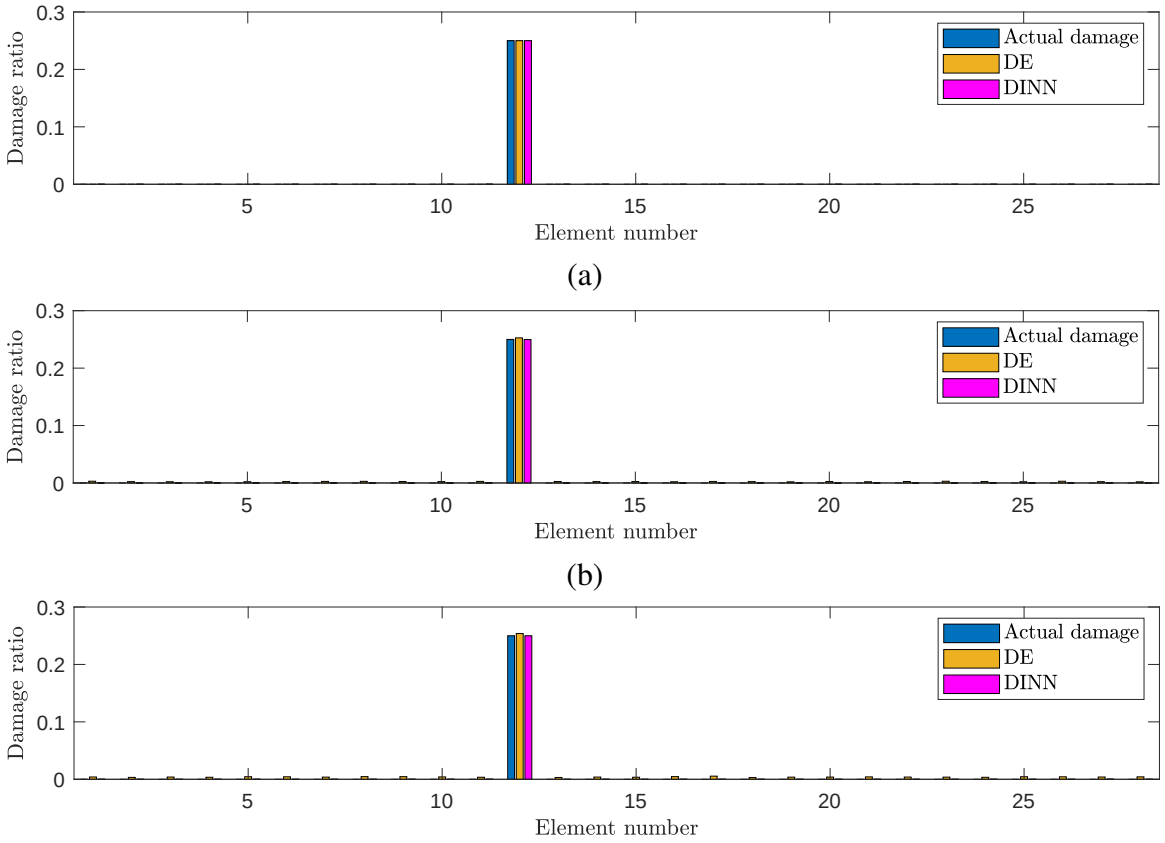
Table 21 reports the optimal hyperparameters found by the BO for all damage scenarios. To evaluate the performance of the present method, Table 22, Figs. 21 and 22 provide a comparison of damage assessment results found by the DINN and other algorithms. As seen in the graphical results, both algorithms correctly detect all damaged members of the frame structure for various noise conditions. From the data of Table 22, it is easily seen that our model outperforms the CS [14] and Jaya [14] in terms of the solution quality and needed analyses. Although the accuracy of damage localization found by the DINN is similar for the DE algorithm, its simulation number (500) dramatically reduces more than 160 times compared with the DE (84,933). Additionally, the mean damage ratios are quite close to the upper and lower bounds of the 95% CI. These results have demonstrated the efficiency and robustness of the DINN for the structural damage identification problem.

Table 21
Optimum hyperparameters of the network obtained using the BO with various damage scenarios for the four-storey structure.

Case	Noise level	Hyperparameters					
		Hidden layers	Hidden neurons	Activation function	Learning rate	Step size	Gamma
1	0%	2	25	LeakyReLU	0.0532	8	0.4441
	5%	3	34	LeakyReLU	0.0114	10	0.4845
	8%	3	36	ReLU	0.0539	8	0.5988
2	0%	4	36	LeakyReLU	0.0277	5	0.8000
	5%	1	20	ReLU	0.0367	8	0.8000
	8%	3	21	LeakyReLU	0.0509	10	0.4679

Table 22
Statistical results of damage assessment from different optimization algorithms for three scenarios of the four-storey structure with various noise levels.

Case	Noise level	Actual Location	CS [14]			Jaya [14]			DE			DINN			
			Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	Mean	Std	Avg	95% CI
1	0%	a_{12}	0.2498	0.0001	58,350	0.2496	0.0001	19,585	0.2501	0.0001	84,933	0.2501	0.0001	500	[0.2501, 0.2502]
	5%	a_{12}	0.2273	0.0206	77,340	0.2278	0.0203	28,240	0.2527	0.0004	150,000	0.2498	0.0002	500	[0.2497, 0.2499]
	8%	a_{12}	0.2208	0.0321	80,000	0.2231	0.0300	29,640	0.2538	0.0008	150,000	0.2499	0.0003	500	[0.2498, 0.2501]
2	0%	a_1	0.1998	0.0001	55,055	0.1997	0.0002	20,840	0.2000	0.0000	94,208	0.1999	0.0001	500	[0.1998, 0.2000]
		a_8	0.2998	0.0001		0.2996	0.0002		0.3000	0.0000		0.3002	0.0001		[0.3001, 0.3003]
		a_{21}	0.1999	0.0001		0.1999	0.0001		0.2000	0.0000		0.2001	0.0002		[0.2001, 0.2002]
	5%	a_1	0.1977	0.0139	69,310	0.1978	0.0138	29,400	0.2025	0.0003	150,000	0.2002	0.0006	500	[0.1997, 0.2006]
		a_8	0.2722	0.0184		0.2722	0.0182		0.3012	0.0002		0.2997	0.0003		[0.2995, 0.2999]
		a_{21}	0.1964	0.0182		0.1964	0.0181		0.2029	0.0003		0.1997	0.0007		[0.1993, 0.2002]
	8%	a_1	0.1899	0.0268	73,780	0.1899	0.0268	31,455	0.2021	0.0004	150,000	0.1994	0.0012	500	[0.1984, 0.2004]
		a_8	0.2531	0.0378		0.2533	0.0377		0.3023	0.0003		0.3002	0.0006		[0.2997, 0.3007]
		a_{21}	0.1770	0.0402		0.1771	0.0403		0.2027	0.0003		0.1995	0.0010		[0.1986, 0.2004]



(a)

(b)

(c)

Fig. 21. The damage identification results of the four-storey structure obtained using the DINN and DE for the scenario 1: (a) noise free; (b) noise level 1; (c) noise level 2.

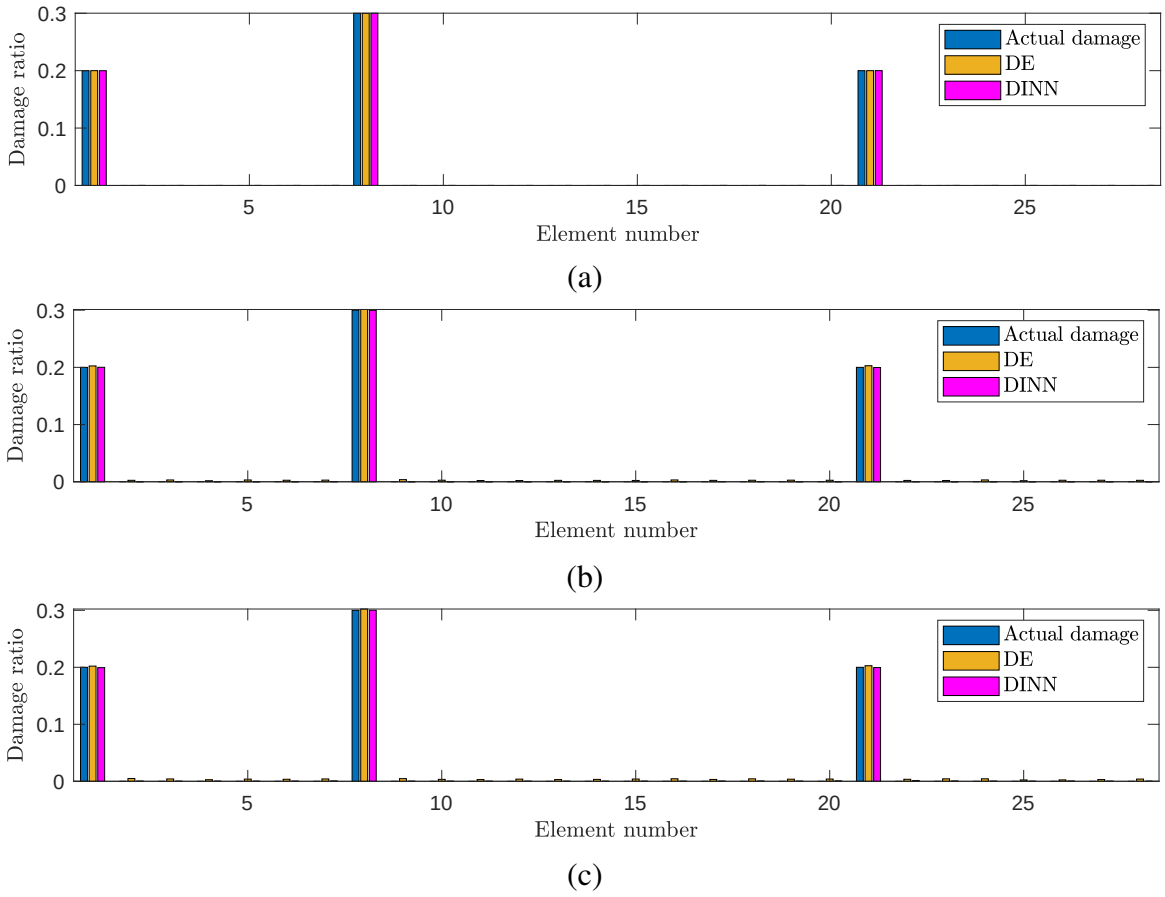


Fig. 22. The damage identification results of the four-storey structure obtained using the DINN and DE for the scenario 2: (a) noise free; (b) noise level 1; (c) noise level 2.

4. Conclusions

In this article, a robust damage-informed NN framework is successfully developed for damage identification of structure. The DNN is designed to parameterize the damage ratios of members, enabling the damage assessment to be converted to an optimization problem. The network is used to minimize a hybrid objective function that includes terms related to the multiple damage location assurance criterion and the flexibility matrices of the modal test and the intact model. And the parameters of the network are immediately pointed out when the training process ends instead of locating the damage ratios. Besides, BO is applied to automatically tune the hyperparameters to determine the optimal network. The applicability, efficiency, and reliability of the proposed paradigm are verified and demonstrated by several numerical examples for structural damage assessment. And the numerical results revealed that our approach can accurately detect the damaged locations of truss and frame structures with the small rel-

ative errors while requiring fewer FEAs than other competitive algorithms. Furthermore, the automatic hyperparameter tuning of DNN based on BO also ensure the high reliability even under noisy modal data and preventing the network from getting trapped in a local minimum. Based on the outstanding features mentioned above, DINN shows great potential as a viable alternative for addressing complex problems.

CRediT authorship contribution statement

Hau T. Mai: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing review & editing, Visualization. **Seunghye Lee**: Formal analysis, Data curation. **Joowon Kang**: Data curation, Validation. **Jaehong Lee**: Conceptualization, Methodology, Supervision, Funding acquisition.

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.

Data availability

The authors do not have permission to share data.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (NRF-2023R1A2C2003310).

References

[1] S. W. Doebling, C. R. Farrar, M. B. Prime, et al., A summary review of vibration-based damage identification methods, Shock and vibration digest 30 (1998) 91–105.

[2] W. Fan, P. Qiao, Vibration-based damage identification methods: a review and comparative study, Structural health monitoring 10 (2011) 83–111.

[3] P. Cawley, R. D. Adams, The location of defects in structures from measurements of natural frequencies, The Journal of Strain Analysis for Engineering Design 14 (1979) 49–57.

[4] S. S. Kessler, S. M. Spearing, M. J. Atalla, C. E. Cesnik, C. Soutis, Damage detection in composite materials using frequency response methods, Composites Part B: Engineering 33 (2002) 87–95.

[5] S. Chinchalkar, Determination of crack location in beams using natural frequencies, Journal of Sound and vibration 247 (2001) 417–429.

[6] M. Cerri, F. Vestroni, Detection of damage in beams subjected to diffused cracking, Journal of Sound and Vibration 234 (2000) 259–276.

[7] E. P. Carden, P. Fanning, Vibration based condition monitoring: a review, Structural health monitoring 3 (2004) 355–377.

[8] J.-T. Kim, Y.-S. Ryu, H.-M. Cho, N. Stubbs, Damage identification in beam-type structures: frequency-based method vs mode-shape-based method, Engineering structures 25 (2003) 57–67.

[9] N. Stubbs, J. Kim, K. Topole, The effect of model uncertainty on the accuracy of global nondestructive damage detection in structures, Computational stochastic mechanics (1991) 157–168.

[10] J. A. dos Santos, C. M. Soares, C. M. Soares, H. Pina, A damage identification numerical model based on the sensitivity of orthogonality conditions and least squares techniques, Computers & Structures 78 (2000) 283–291.

[11] Z. Shi, S. Law, L. Zhang, Damage localization by directly using incomplete mode shapes, Journal of engineering mechanics 126 (2000) 656–660.

[12] M. A. Wahab, Effect of modal curvatures on damage detection using model updating, Mechanical Systems and Signal Processing 15 (2001) 439–445.

[13] R. Sampaio, N. Maia, J. Silva, Damage detection using the frequency-response-function curvature method, Journal of sound and vibration 226 (1999) 1029–1042.

[14] D.-C. Du, H.-H. Vinh, V.-D. Trung, N.-T. Hong Quyen, N.-T. Trung, Efficiency of jaya algorithm for solving the optimization-based structural damage identification problem based on a hybrid objective function, Engineering Optimization 50 (2018) 1233–1251.

[15] A. Pandey, M. Biswas, Damage detection in structures using changes in flexibility, Journal of sound and vibration 169 (1994) 3–17.

[16] G.-Q. Li, K.-C. Hao, Y. Lu, S.-W. Chen, A flexibility approach for damage identification of cantilever-type structures with bending and shear deformation, Computers & structures 73 (1999) 565–572.

[17] L. Stutz, D. Castello, F. Rochinha, A flexibility-based continuum damage identification approach, Journal of Sound and Vibration 279 (2005) 641–667.

[18] A. Z. Hosseinzadeh, A. Bagheri, G. G. Amiri, K.-Y. Koo, A flexibility-based method via the iterated improved reduction system and the cuckoo optimization algorithm for damage quantification with limited sensors, Smart Materials and Structures 23 (2014) 045019.

[19] M. I. Friswell, Damage identification using inverse methods, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 365 (2007) 393–410.

[20] N. F. Alkayem, M. Cao, Y. Zhang, M. Bayat, Z. Su, Structural damage detection using finite element model updating with evolutionary algorithms: a survey, Neural Computing and Applications 30 (2018) 389–411.

[21] A. Teughels, G. De Roeck, J. A. Suykens, Global optimization by coupled local minimizers and its application to fe model updating, Computers & structures 81 (2003) 2337–2351.

[22] P. G. Bakir, E. Reynders, G. De Roeck, An improved finite element model updating

596 method by the global optimization technique 'coupled local minimizers', Computers &
597 Structures 86 (2008) 1339–1352.

598 [23] S.-E. Fang, R. Perera, G. De Roeck, Damage identification of a reinforced concrete frame
599 by finite element model updating using damage parameterization, Journal of Sound and
600 Vibration 313 (2008) 544–559.

601 [24] W.-X. Ren, H.-B. Chen, Finite element model updating in structural dynamics by using
602 the response surface method, Engineering structures 32 (2010) 2455–2465.

603 [25] Y.-Q. Wang, L. Zong, Y.-J. Shi, N. Yao, Damage detection and rehabilitation on a curvi-
604 linear steel box girder bridge by multistage model updating, Structure and Infrastructure
605 Engineering 11 (2015) 1420–1431.

606 [26] Y. Zhang, C. Jia, J. Li, B. Spencer Jr, Model updating based on an affine scaling interior
607 optimization algorithm, Engineering Optimization 45 (2013) 1379–1395.

608 [27] T. Vo-Duy, V. Ho-Huu, H. Dang-Trung, T. Nguyen-Thoi, A two-step approach for damage
609 detection in laminated composite structures using modal strain energy method and an
610 improved differential evolution algorithm, Composite Structures 147 (2016) 42–53.

611 [28] R. Perera, R. Torres, Structural damage detection via modal data with genetic algorithms,
612 Journal of Structural Engineering 132 (2006) 1491–1501.

613 [29] F. Kang, J.-j. Li, Q. Xu, Damage detection based on improved particle swarm optimization
614 using vibration data, Applied Soft Computing 12 (2012) 2329–2335.

615 [30] H. Xu, J. Liu, Z. Lu, Structural damage identification based on cuckoo search algorithm,
616 Advances in Structural Engineering 19 (2016) 849–859.

617 [31] Z. Ding, K. Fu, W. Deng, J. Li, L. Zhongrong, A modified artificial bee colony algorithm
618 for structural damage identification under varying temperature based on a novel objective
619 function, Applied Mathematical Modelling 88 (2020) 122–141.

[32] R. D. Kundu, M. Mishra, D. Maity, Teaching–learning-based optimization algorithm for solving structural damage detection problem in frames via changes in vibration responses, Architecture, Structures and Construction (2021) 1–20.

[33] Q. X. Lieu, V. H. Luong, J. Lee, Structural damage identification using adaptive hybrid evolutionary firefly algorithm, Applications of firefly algorithm and its variants: case studies and new developments (2020) 75–97.

[34] H.-Y. Fan, J. Lampinen, A trigonometric mutation operation to differential evolution, Journal of global optimization 27 (2003) 105–129.

[35] H. T. Mai, J. Kang, J. Lee, A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior, Finite Elements in Analysis and Design 196 (2021) 103572.

[36] H. T. Mai, S. Lee, D. Kim, J. Lee, J. Kang, J. Lee, Optimum design of nonlinear structures via deep neural network-based parameterization framework, European Journal of Mechanics-A/Solids (2022) 104869.

[37] H. T. Mai, Q. X. Lieu, J. Kang, J. Lee, A novel deep unsupervised learning-based framework for optimization of truss structures, Engineering with Computers (2022) 1–24.

[38] H. T. Mai, D. D. Mai, J. Kang, J. Lee, J. Lee, Physics-informed neural energy-force network: a unified solver-free numerical simulation for structural optimization, Engineering with Computers (2023) 1–24.

[39] A. Chandrasekhar, K. Suresh, Tounn: Topology optimization using neural networks, Structural and Multidisciplinary Optimization 63 (2021) 1135–1149.

[40] H. T. Mai, T. T. Truong, J. Kang, D. D. Mai, J. Lee, A robust physics-informed neural network approach for predicting structural instability, Finite Elements in Analysis and Design 216 (2023) 103893.

[41] H. T. Mai, Q. X. Lieu, J. Kang, J. Lee, A robust unsupervised neural network framework

for geometrically nonlinear analysis of inelastic truss structures, Applied Mathematical Modelling 107 (2022) 332–352.

[42] W. Li, M. Z. Bazant, J. Zhu, A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches, Computer Methods in Applied Mechanics and Engineering 383 (2021) 113933.

[43] Q. X. Lieu, K. T. Nguyen, K. D. Dang, S. Lee, J. Kang, J. Lee, An adaptive surrogate model to structural reliability analysis using deep neural network, Expert Systems with Applications 189 (2022) 116104.

[44] X. Wu, J. Ghaboussi, J. Garrett Jr, Use of neural networks in detection of structural damage, Computers & structures 42 (1992) 649–659.

[45] P. Tsou, M.-H. H. Shen, Structural damage detection and identification using neural networks, AIAA journal 32 (1994) 176–183.

[46] Y. Teboub, P. Hajela, A neural network based damage analysis of smart composite beams, in: 4th Symposium on Multidisciplinary Analysis and Optimization, 1992, p. 4685.

[47] M. Mehrjoo, N. Khaji, H. Moharrami, A. Bahreininejad, Damage detection of truss bridge joints using artificial neural networks, Expert systems with applications 35 (2008) 1122–1131.

[48] T. T. Truong, D. Dinh-Cong, J. Lee, T. Nguyen-Thoi, An effective deep feedforward neural networks (dfnn) method for damage identification of truss structures using noisy incomplete modal data, Journal of Building Engineering 30 (2020) 101244.

[49] F. Parisi, A. Mangini, M. Fanti, J. M. Adam, Automated location of steel truss bridge damage using machine learning and raw strain sensor data, Automation in Construction 138 (2022) 104249.

[50] N. Wang, X. Zhao, P. Zhao, Y. Zhang, Z. Zou, J. Ou, Automatic damage detection of historic masonry buildings based on mobile deep learning, Automation in Construction 103 (2019) 53–66.

[51] X. Cui, Q. Wang, S. Li, J. Dai, C. Xie, Y. Duan, J. Wang, Deep learning for intelligent identification of concrete wind-erosion damage, Automation in Construction 141 (2022) 104427.

[52] L. V. Ho, D. H. Nguyen, M. Mousavi, G. De Roeck, T. Bui-Tien, A. H. Gandomi, M. A. Wahab, A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks, Computers & Structures 252 (2021) 106568.

[53] D. W. Abueidda, S. Koric, E. Guleryuz, N. A. Sobh, Enhanced physics-informed neural networks for hyperelasticity, International Journal for Numerical Methods in Engineering 124 (2023) 1585–1601.

[54] L. Linden, D. K. Klein, K. A. Kalina, J. Brummund, O. Weeger, M. Kästner, Neural networks meet hyperelasticity: A guide to enforcing physics, Journal of the Mechanics and Physics of Solids (2023) 105363.

[55] D. Chakraborty, Artificial neural network based delamination prediction in laminated composites, Materials & design 26 (2005) 1–7.

[56] J.-H. Bastek, D. M. Kochmann, Physics-informed neural networks for shell structures, European Journal of Mechanics-A/Solids 97 (2023) 104849.

[57] Z. Mao, A. D. Jagtap, G. E. Karniadakis, Physics-informed neural networks for high-speed flows, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112789.

[58] H. Jeong, C. Batuwatta-Gamage, J. Bai, Y. M. Xie, C. Rathnayaka, Y. Zhou, Y. Gu, A complete physics-informed neural network-based framework for structural topology optimization, Computer Methods in Applied Mechanics and Engineering 417 (2023) 116401.

[59] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, et al., Jax: composable transformations of python+ numpy programs (2018).

[60] A. Messina, E. Williams, T. Contursi, Structural damage detection by a sensitivity and statistical-based method, Journal of sound and vibration 216 (1998) 791–808.

[61] J. Zhao, J. T. DeWolf, Sensitivity study for vibrational parameters used in damage detection, Journal of structural engineering 125 (1999) 410–416.

[62] F. Hutter, L. Kotthoff, J. Vanschoren, Automated machine learning: methods, systems, challenges, Springer Nature, 2019.

[63] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, S.-H. Deng, Hyperparameter optimization for machine learning models based on bayesian optimization, Journal of Electronic Science and Technology 17 (2019) 26–40.

[64] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (1998) 455.

[65] A. Chandrasekhar, S. Sridhara, K. Suresh, Auto: a framework for automatic differentiation in topology optimization, Structural and Multidisciplinary Optimization 64 (2021) 4355–4365.

[66] D. Dinh-Cong, T. Vo-Duy, T. Nguyen-Thoi, Damage assessment in truss structures with limited sensors using a two-stage method and model reduction, Applied Soft Computing 66 (2018) 264–277.

[67] V. Ho-Huu, T. Vo-Duy, T. Luu-Van, L. Le-Anh, T. Nguyen-Thoi, Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme, Automation in Construction 68 (2016) 81–94.

[68] N. Fallah, S. R. H. Vaez, A. Mohammadzadeh, Multi-damage identification of large-scale truss structures using a two-step approach, Journal of Building Engineering 19 (2018) 494–505.