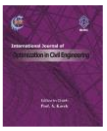
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2025; 15(4):511-526



PREDICTING THE NATURAL FREQUENCIES OF TRUSS DOMES UNDER UNCERTAINTY USING DEEP FEEDFORWARD NEURAL NETWORKS

Pooya Zakian[*,†,1,2] and Pegah Zakian[3]
[1]*Department of Civil Engineering, Faculty of Engineering, Arak University, Arak, Iran*
[2]*Faculty of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran*
[3]*Department of Electrical Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran*

[DOI: 10.22068/ijoce.2025.15.4.652]

ABSTRACT

This study employs Monte Carlo simulation together with a deep feedforward neural network to predict the natural frequencies of truss domes under uncertainty. Material and/or geometric properties of these structures are modeled as random variables, and their influence on the natural frequencies is examined. Monte Carlo simulation is applied to perform stochastic eigenvalue analyses of the finite element models. To reduce computational cost, a deep neural network is trained to predict natural frequencies in place of repeated eigenvalue solves, accelerating the overall simulation. Bayesian optimization is used to tune the network hyperparameters. Numerical examples show that the proposed approach substantially improves computational efficiency and predictive accuracy compared with direct Monte Carlo simulation for domes with random inputs.

Keywords: Machine learning; feedforward neural network; Dome; Random eigenvalue problem; Bayesian optimization; Monte Carlo simulation.

Received: 11 September 2025; Accepted: 2 November 2025

1. INTRODUCTION

Uncertainty plays a significant role in the analysis and design of structural systems, making probabilistic approaches essential. As a result, numerous techniques have been developed to

[*]Corresponding author: Department of Civil Engineering, Faculty of Engineering, Arak University, Arak, Iran
[†]E-mail address: p-zakian@araku.ac.ir (P. Zakian)

quantify uncertainties in structural models [1-3] . Among these, Monte Carlo simulation (MCS) is a widely recognized method for probabilistic analysis. It operates as a sampling technique, generating numerous realizations based on randomly selected values for uncertain parameters. MCS has been extensively applied in reliability assessments of structures [4, 5]. Despite its robustness, MCS is computationally intensive, as each realization requires a full structural analysis, which can be time-consuming. To address this limitation, alternative sampling strategies or surrogate models (such as those based on machine learning) can be employed to reduce computational demands. Furthermore, techniques like Latin hypercube sampling or importance sampling can lower the required number of samples, while machine learning models serve as efficient surrogates for the original structural model.

With the rapid advancement of computer technology, machine learning has found widespread application across various domains. Today, machine learning techniques are extensively applied across fields such as computational mechanics [6, 7]; material modeling [8]; structural optimization [9, 10]; error-resilient system design [11-13]; and electric power system analysis [14]. Nevertheless, the primary application areas of machine learning in structural mechanics include modeling of structural materials, seismic response prediction, wind-load estimation, structural health monitoring, structural optimization, damage detection and localization, and structural control [8, 9, 15-18].

In recent years, deep neural network surrogates have been developed to reduce the computational burden of finite element analysis [8, 19, 20]. For example, Mai et al. [21] trained a deep neural network on data from finite element analysis and integrated it with a differential evolution algorithm to efficiently optimize geometrically nonlinear space trusses under displacement constraints. In a separate study [22], a robust framework was proposed that parameterizes truss cross-sectional areas through weights and biases of a deep neural network—using joint coordinates as inputs—and embeds this surrogate within a Bayesian optimization loop to directly identify minimum-weight designs under displacement constraints.

Representing uncertainty in natural frequencies is essential for analyzing the dynamic behavior of structural systems [23]. When adequate statistical data is available, stochastic approaches are commonly employed to account for uncertainties. These methods typically model uncertainties as random variables, stochastic processes, or random fields. The characterization of uncertainty is often expressed through parameters such as probability density functions, mean values, and variances. However, in cases where the probability density function is unknown or insufficiently defined, uncertainty can alternatively be represented using interval bounds that specify the upper and lower limits of the random variables.

The natural frequencies of structures with uncertain parameters are typically determined by solving random eigenvalue problems. Scheidt and Purker [24] conducted foundational research in this area. Several methodologies have been developed to address these problems, including the direct MCS approach [25], and the perturbation method [23]. Hollot and Bartlett [26] explored the eigenvalues of interval matrices, while Chen et al. [27] introduced perturbation techniques for estimating the bounds of eigenvalues in vibrating systems with interval-based parameters. Qiu et al. [28] applied the vertex theorem to compute eigenvalue bounds for structures characterized by uncertain-but-bounded parameters. Gao [29] proposed the interval factor method for analyzing the natural frequencies and mode shapes of structures

[DOI: 10.22068/ijoce.2025.15.4.652]

with interval uncertainties. Modares et al. [30] developed an element-by-element formulation to address interval eigenvalue problems, and Angeli et al. [31] investigated frequency intervals in systems exhibiting polytopic uncertainty. Recently, support vector machines were employed in MCS to accelerate the computation of eigenvalues for truss structures with uncertain parameters [32]. Furthermore, numerous studies have focused on stochastic modeling to evaluate the dynamic behavior of structures under uncertainty. However, practical structural systems often involve a large number of variables and design parameters, some of which possess adequate statistical data while others do not. Consequently, a combined use of stochastic and interval models becomes necessary. In this context, various efforts have been made to solve mixed stochastic problems in both static [33, 34] and dynamic [35] analyses.

In this study, MCS is employed to compute the natural frequencies of truss domes under uncertainty. A set of random variables is included in the MCS to perform a stochastic eigenvalue analysis of the system. To enhance simulation efficiency, a deep feedforward neural network (DFNN) is trained to predict natural frequencies, enabling faster execution of each MCS iteration. This network is used as a surrogate model, replacing the finite element eigenvalue analysis of the structure. We perform hyperparameter tuning on this network, for which Bayesian optimization is used to identify the best settings. The proposed approach is evaluated on two examples, demonstrating its computational efficiency and accuracy compared with direct MCS.

The rest of this paper is organized as follows. Section 2 reviews the research background of deep neural networks. Section 3 outlines eigenvalue analysis for determining the natural frequencies of truss structures and introduces the proposed approach. Section 4 presents illustrative examples, and Section 5 offers the conclusions.

2. DEEP FEEDFORWARD NEURAL NETWORK (DFNN)

A feedforward neural network, also known as a fully connected neural network, is among the earliest models developed in artificial intelligence [36]. The multilayer perceptron (MLP) is a variant of the feedforward neural network. It consists of three primary layers: the input layer, the hidden layer, and the output layer, as illustrated in Figure 1. The input layer receives the incoming signal, while the output layer performs tasks such as classification and prediction. Between these two layers lies a series of hidden layers, which may extend indefinitely and are fully connected. In an MLP, data flows unidirectionally from the input to the output layer, following the feedforward architecture. The network is trained using the backpropagation algorithm, which adjusts the weights of all nodes. MLPs are capable of solving problems that are not linearly separable and are designed to approximate any continuous function. Multi-layer neural networks establish mathematical mappings between inputs and outputs by iteratively tuning weights and biases during training. A network with a single hidden layer is termed shallow, while architectures with two or more hidden layers are known as deep neural networks. Figure 1 depicts the fully connected architecture of the network, comprising an input layer, $(N-1)$ hidden layers, and an output layer. Each neuron in layer (i) receives signals from all neurons in layer $(i-1)$ via a weight matrix $\mathbf{W}^{(i)}$ and a bias vector $\mathbf{b}^{(i)}$. The layer outputs are computed as

[DOI: 10.22068/ijoce.2025.15.4.652]

$$\begin{aligned}\mathbf{h}^{(i)} &= f_I\left(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}\right), \qquad \forall\, i = 1,2,...,(N-1)\\ \hat{\mathbf{y}} &= f_O\left(\mathbf{W}^{(N)}\mathbf{h}^{(N-1)} + \mathbf{b}^{(N)}\right),\end{aligned} \tag{1}$$

where f_I and f_O are the activation functions for the hidden and output layers, respectively; $\hat{\mathbf{y}}$ represents the output vector, while the input vector is denoted as $\mathbf{h}^{(0)} = \mathbf{x}$.

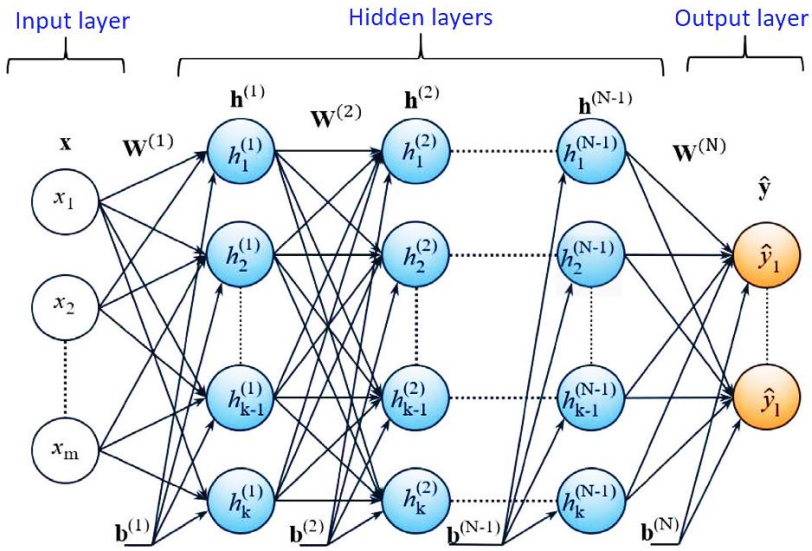


Figure 1: Schematic representation of a deep feedforward neural network [21].

In this section, a surrogate model is developed using a DFNN to predict eigenvalues of dome structures, so that eigenvalue analysis is replaced by the trained DFNN. Therefore, MCS with the DFNN performs much faster than MCS using the actual eigenvalue analysis, while providing an approximate solution. The three main components of generating the surrogate model using the DFNN are summarized as follows:

(1) A set of samples is generated using Latin Hypercube Sampling for the uncertain parameters, such as Young's modulus, mass density, and cross-sectional areas. An eigenvalue analysis is carried out for each sample to collect the required natural frequency responses. All input and output variables are normalized before being fed into the neural network.
(2) A DFNN is developed and trained to approximate the relationship between the inputs and the natural frequencies.
(3) The eigenvalue analysis is replaced by the DFNN surrogate to predict the natural frequencies.

In supervised learning, labeled data guide the optimization of weights and biases by minimizing a loss function. For regression tasks, the mean squared error (MSE) is widely used, defined as

[DOI: 10.22068/ijoce.2025.15.4.652]

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2 \quad (2)$$

where y_j and $\hat{y}_j$ denote the observed (true) and predicted outputs for the *j*th data point, and *n* is the total number of scalar predictions, i.e., the number of samples multiplied by the number of output units.

Before training, inputs and targets (outputs) are standardized using z-score normalization. This normalization procedure expresses each value's distance from the mean in units of the standard deviation, producing a dataset with mean 0 and standard deviation 1 while preserving the original distribution's shape, skewness, and kurtosis. Standardization improves optimization stability and accelerates convergence. To further speed training and reduce variance in gradient estimates, the dataset is split into mini-batches and parameters are updated batch-wise using mini-batch gradient descent rather than using the full dataset or individual samples.

Choosing appropriate activation functions, commonly rectified linear unit (ReLU) or sigmoid, allows the network to model nonlinear relationships and improves regression accuracy. In this study, each hidden block contains three layers: a fully connected layer, a batch normalization layer, and a ReLU activation. The ReLU applies a nonlinear threshold by setting negative inputs to zero and leaving positive inputs unchanged.

Backpropagation uses the gradient of the loss function to iteratively update every weight and bias. Over the years, researchers have proposed optimizers such as SGD, Adagrad, Adadelta, and RMSprop. Adam, which combines Adagrad-style per-parameter learning rates with RMSprop-style momentum, has emerged as a robust choice for training on nonconvex problems [21, 37]. Recent studies demonstrate deep learning's effectiveness in structural analysis and optimization [8, 22, 38], so we adopt Adam to train our DFNN on the available data.

Optimization generally seeks the point that minimizes or maximizes a real-valued objective function. Bayesian optimization is one such approach. It builds and updates a Gaussian process surrogate of the objective using past function evaluations. An acquisition function then guides the choice of the next evaluation by balancing exploitation of regions with low predicted objective values and exploration of poorly modeled regions. Bayesian optimization is widely used for hyperparameter tuning of machine learning algorithms [39]. In this paper, the DFNN hyperparameters are tuned using Bayesian optimization to minimize the MSE under the given parameter bounds.

3. ANALYSIS OF VIBRATION FREQUENCIES

In this section, first, the vibration frequency analysis of truss structures is reviewed, followed by a discussion of probabilistic analysis using Monte Carlo simulation and deep neural networks.

[DOI: 10.22068/ijoce.2025.15.4.652]

3.1. Eigenvalue analysis

To compute a structure's natural frequencies, one must solve the eigenvalue problem involving its stiffness and mass matrices [40-42]. The stiffness matrix for a three-dimensional truss element is then given by:

$$\mathbf{k}^e = \frac{EA}{L}\begin{bmatrix} C_x^2 & C_xC_y & C_xC_z & -C_x^2 & -C_xC_y & -C_xC_z \\ & C_y^2 & C_yC_z & -C_xC_y & -C_y^2 & -C_yC_z \\ & & C_z^2 & -C_xC_z & -C_yC_z & -C_z^2 \\ & & & C_x^2 & C_xC_y & C_xC_z \\ & & & & C_y^2 & C_yC_z \\ \textit{Symmetric} & & & & & C_z^2 \end{bmatrix} \tag{3}$$

with

$$C_x = \frac{x_j - x_i}{L}, \quad C_y = \frac{y_j - y_i}{L}, \quad C_z = \frac{z_j - z_i}{L} \tag{4}$$

where $\mathbf{k}^e$ denotes the stiffness matrix of the truss element connecting nodes *i* and *j*; L, A and E are the element length, cross- sectional area, and Young's modulus, respectively; x_i , y_i and z_i are the Cartesian coordinates of node *i*. The element stiffness matrices are assembled into $\mathbf{K}$.

The consistent mass matrix for a three-dimensional truss element is expressed as

$$\mathbf{m}^e = \frac{\rho AL}{6}\begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix} \tag{5}$$

in which ρ represents the material density. The element mass matrices are assembled into $\mathbf{M}_{\text{struct}}$, which contains the structural mass, while nonstructural masses, if present, are represented by a lumped mass matrix denoted as $\mathbf{M}_{\text{nonstruct}}$; thus the total mass matrix is $\mathbf{M} = \mathbf{M}_{\text{struct}} + \mathbf{M}_{\text{nonstruct}}$, and if no nonstructural masses exist then $\mathbf{M} = \mathbf{M}_{\text{struct}}$.

Using the assembled stiffness and mass matrices, the global eigenvalue problem becomes:

$$\mathbf{K}\phi_k = \omega_k^2 \mathbf{M}\phi_k \tag{6}$$

[DOI: 10.22068/ijoce.2025.15.4.652]

where **M** and **K** denote the mass and stiffness matrices of the structure, respectively; ω_k and ϕ_k represent the kth circular frequency and its corresponding mode shape vector, respectively.

3.2. The MCS-DFNN procedure

To estimate the statistical measures of uncertain structural responses, MCS employs repeated random sampling based on prescribed probability distributions. In this approach, each uncertain variable is modeled by a probability distribution, and the simulation recalculates the output repeatedly, using a new set of random samples each time. Three key steps define the MCS workflow:

- Identify the predictive model by specifying independent variables (random inputs) and the dependent variable (response of interest).
- Assign probability distributions to the independent variables, drawing on historical data or expert judgment, then generate random samples accordingly.
- Run simulations for a predefined number of samples—solving, for example, a deterministic eigenvalue problem for each realization—until the desired accuracy of response statistics (e.g., mean, standard deviation) is achieved.

Although MCS is straightforward and robust for stochastic structural mechanics, its accuracy hinges on the number of samples: the standard deviation error decreases with the square root of the sample size. Consequently, large-scale systems with many random parameters incur high computational costs due to the sheer volume of deterministic solves. To alleviate this burden, we replace the direct eigenvalue analyses with a DFNN surrogate, yielding an MCS-DFNN framework. The procedure unfolds as follows:

- Construct a dataset of observations pairing input random variables (features) with their resulting natural frequencies (targets). Each training sample represents one set of random structural parameters and the corresponding frequency.
- Train a DFNN regression model on this dataset to learn the mapping from input parameters to natural frequencies.
- Perform MCS without invoking structural eigenvalue solvers in each iteration. Instead, feed each random sample into the trained DFNN to predict natural frequencies, dramatically reducing per-sample computational cost.

This MCS-DFNN approach maintains the statistical rigor of conventional MCS while achieving substantial speed-ups for stochastic eigenvalue analysis of the dome structures.

4. ILLUSTRATIVE EXAMPLES

Two truss-dome examples are analyzed to demonstrate the accuracy and efficiency of the proposed MCS-DFNN method for computing natural frequencies of lower vibration modes. For each example, results from the direct MCS and the proposed MCS-DFNN approach are

[DOI: 10.22068/ijoce.2025.15.4.652]

compared in terms of solution accuracy and computational cost. In the MCS approach, an eigenvalue analysis is performed for every sample, whereas the MCS-DFNN replaces the per-sample eigenvalue computation with predictions from a DFNN. The MCS is carried out using a total of 20,000 samples. For DFNN training, the dataset of size N is randomly partitioned into training (70%), validation (15%), and testing (15%) subsets; data points are generated by eigenvalue analyses of the structure under different realizations of the random variables. The LHS was utilized to create efficiently distributed data points.

Hyperparameter tuning is performed for each example using Bayesian optimization over 60 objective-function evaluations to optimize the number of hidden layers, number of neurons per hidden layer, initial learning rate, decay rate for first-moment (mean) of gradients, decay rate for second-moment (squared gradients), and L2 regularization coefficient (weight decay). The search ranges for these variables are reported in Table 1. The fixed training hyperparameters are set as maximum number of epochs = 80, minimum batch size = 64, learning-rate drop factor = 0.1, and learning-rate drop period = 20. The model is trained using the Adam optimizer. All computations are carried out on a laptop with an Intel Core i7-7700HQ CPU at 2.80 GHz and 16 GB of RAM.

Table 1: Range of hyperparameter values used for tuning the deep feedforward neural network.

Variable	Type	Range
Number of hidden layers	Integer	[1, 5]
Number of neurons per hidden layer	Integer	[8, 64]
Initial learning rate	Real	$[10^{-4}, 10^{-2}]$
Gradient decay factor	Real	[0.8, 0.95]
Squared gradient decay factor	Real	[0.98, 0.999]
L2 regularization coefficient	Real	$[10^{-5}, 10^{-3}]$

4.1. A 600-bar dome

A single-layer truss dome from Refs. [32, 43] is shown in Figure 2. The dome has a span of 28 m and a height of 7.5 m. It comprises 216 nodes and 600 members formed by cyclic replication of a 9-node, 25-member substructure; the angular increment between consecutive substructures is 15°, producing 24 identical sectors. The mean cross-sectional area, Young's modulus, and material density for all members are 2×10^{-3} m^2, 2×10^5 MPa, and 7850 kg/m^3, respectively. All ground-level nodes are simply supported. Cross-sectional area, Young's modulus, and mass density are modeled as identically distributed normal random variables across members with coefficients of variation 0.16, 0.1, and 0.1, respectively

Six natural frequencies are required. Thus, a dataset of size 1000 is generated using eigenvalue analyses corresponding to 1000 samples of the dome structure; each sample includes three input features (cross-sectional area, Young's modulus, and mass density) and six outputs. Using Bayesian optimization, a DFNN is trained 60 times with varying hyperparameters to find the best set from the list of optimization variables given in Table 1. The optimized parameters are provided in Table 2, and the convergence curves are shown in Figure 3. One curve plots the observed values, i.e., the best objective value actually obtained by evaluating the true objective to date (the incumbent minimum from real function evaluations). The other curve plots the estimated values, i.e., the optimizer's current belief

[DOI: 10.22068/ijoce.2025.15.4.652]

about the minimum based on the Gaussian process surrogate (the surrogate's predicted best objective derived from the Gaussian process posterior and the acquisition strategy).

True (observed) and predicted values for the first six natural frequencies are plotted in Figure 4, demonstrating that these values closely match and that the trained DFNN performs well. The R^2 and *RMSE* values are also shown in Figure 4 to indicate the solution accuracy. Furthermore, Table 3 lists the means, standard deviations, and computational times for the MCS and MCS-DFNN approaches. The MCS-DFNN method is substantially faster than direct MCS, while producing mean and standard-deviation estimates that closely match the MCS results. Table 3 also confirms that errors increase for higher modes, as expected when comparing the frequency accuracy across modes resulting from a trained network.

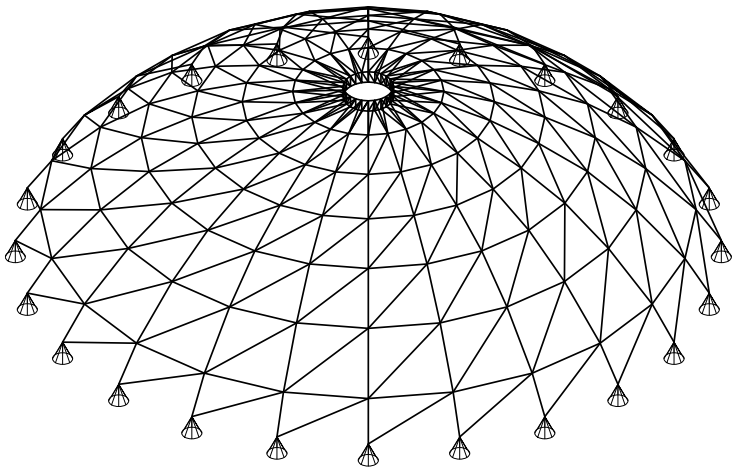


Figure 2: A 600-bar truss dome.

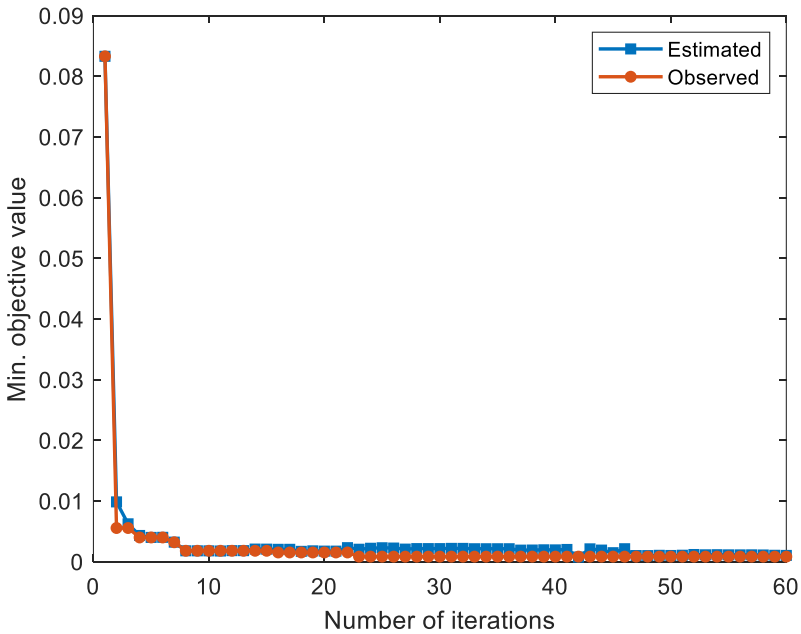


Figure 3: Convergence curves for Bayesian optimization applied to hyperparameter tuning of the deep feedforward neural network for the 600-bar truss dome.

[DOI: 10.22068/ijoce.2025.15.4.652]

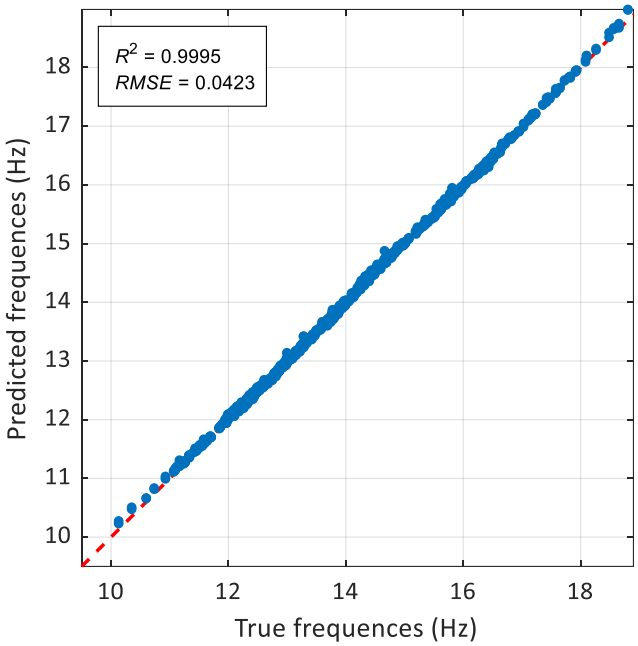


Figure 4: The observed and predicted natural frequencies of the 600-bar truss dome.

Table 2: Optimized hyperparameters of the deep feedforward neural network used to predict the natural frequencies of the 600-bar truss dome.

Variable	Value
Number of hidden layers	4
Number of neurons per hidden layer	62
Initial learning rate	0.0078
Gradient decay factor	0.9449
Squared gradient decay factor	0.9825
L2 regularization coefficient	1.4739×10^{-4}

Table 3: Mean and standard deviation of the natural frequencies for the 600-bar truss dome obtained with MCS and MCS-DFNN.

Natural frequency (Hz)	MCS		MCS-DFNN	
	Mean	Standard deviation	Mean	Standard deviation
f_1	12.5542	0.8957	12.5575	0.8887
f_2	12.5542	0.8957	12.5548	0.8829
f_3	13.5409	0.9661	13.5394	0.9494
f_4	13.5409	0.9661	13.5453	0.9578
f_5	16.1057	1.1491	16.1090	1.1413
f_6	16.1057	1.1491	16.1105	1.1428
Elapsed time (s)	1.3962×10^{3}		4.4012	

4.2. A 1180-bar dome

A truss dome with 400 nodes and 1180 members is considered in this example, as shown in Figure 5. The dome is generated by cyclically replicating a substructure of 20 nodes and 59

[DOI: 10.22068/ijoce.2025.15.4.652]

members, with an 18° angle between consecutive substructures. Each free node has a nonstructural mass of 100 kg. This is a benchmark example in structural optimization [44-46], but here we use it to obtain the first three natural frequencies, treating the 59 members' cross-sectional areas of the substructure as uniformly distributed random variables in the range [0.001, 0.005] m^2. Young's modulus and mass density are deterministic, taken as 2×10^5 MPa and 7850 kg/m^3, respectively. Geometry details and member grouping of the structure can be found in Ref. [46].

A dataset of 6000 samples is generated, each sample containing the 59 cross-sectional area values as inputs and three natural frequencies as targets. Using Bayesian optimization, a DFNN is trained 60 times over the hyperparameter sets listed in Table 1 to identify the best set. The optimized parameters appear in Table 4, and the convergence curves are presented in Figure 6. Figure 7 compares true (observed) and predicted values of the natural frequencies, showing close agreement and desirable DFNN performance. Table 5 reports the means, standard deviations, and computational times for the MCS and MCS-DFNN approaches. The MCS-DFNN approach is substantially faster than direct MCS and yields mean and standard-deviation estimates that largely match the MCS results; however, agreement is less exact than in the previous example because this problem has many more input variables (see also the R^2 and *RMSE* values in Figure 7).

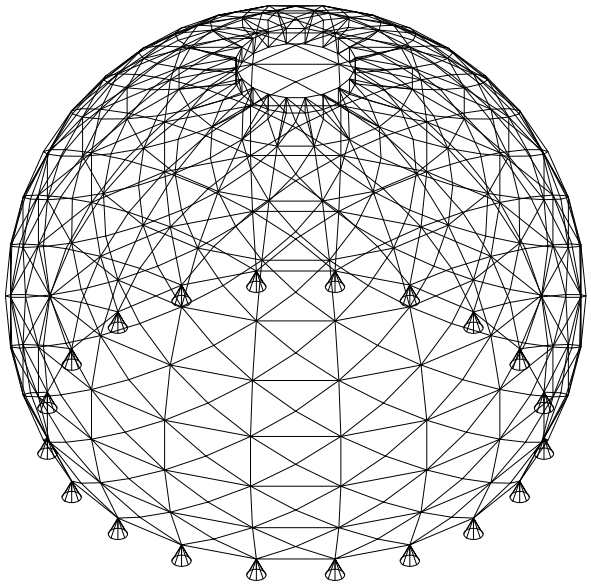


Figure 5: A 1180-bar truss dome.

Table 4: Optimized hyperparameters of the deep feedforward neural network used to predict the natural frequencies of the 1180-bar truss dome.

Variable	Value
Number of hidden layers	5
Number of neurons per hidden layer	64
Initial learning rate	0.0096
Gradient decay factor	0.9239
Squared gradient decay factor	0.9869
L2 regularization coefficient	9.8338×10^{-4}

[DOI: 10.22068/ijoce.2025.15.4.652]

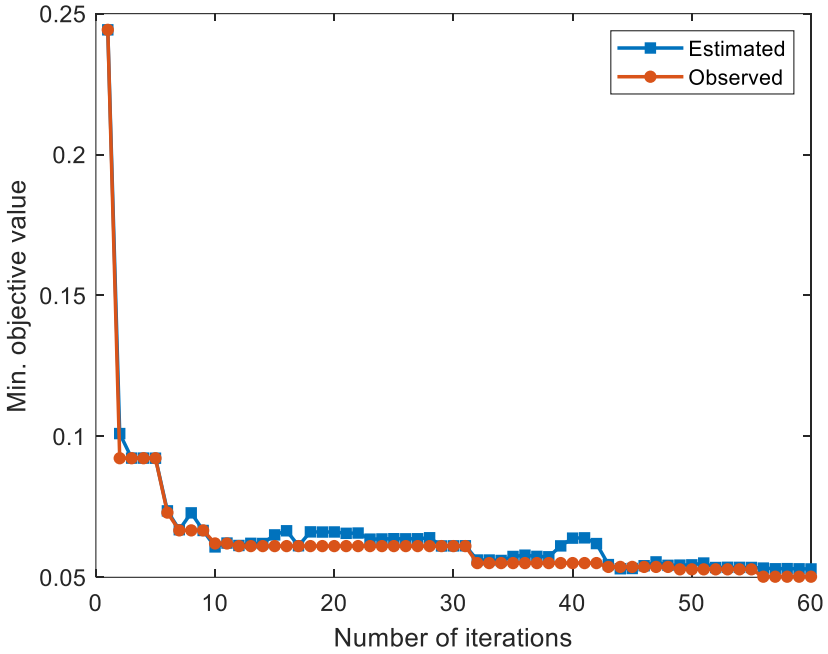


Figure 6: Convergence curves for Bayesian optimization applied to hyperparameter tuning of the deep feedforward neural network for the 1180-bar truss dome.

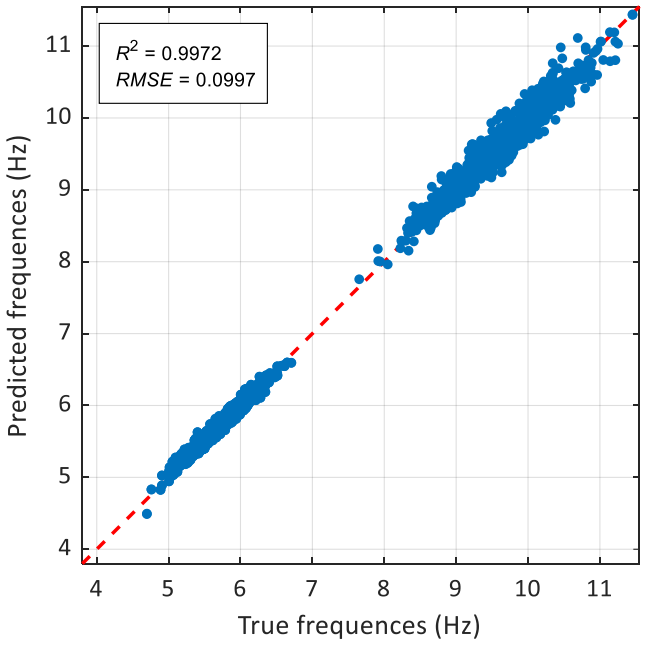


Figure 7: The observed and predicted natural frequencies of the 1180-bar truss dome.

[DOI: 10.22068/ijoce.2025.15.4.652]

Table 5: Mean and standard deviation of the natural frequencies for the 1180-bar truss dome obtained with MCS and MCS-DFNN.

Natural frequency (Hz)	MCS		MCS-DFNN	
	Mean	Standard deviation	Mean	Standard deviation
f_1	5.7493	0.3152	5.7513	0.3056
f_2	5.7493	0.3152	5.7513	0.3058
f_3	9.6177	0.6126	9.6220	0.5893
Elapsed time (s)	3.9284×10^3		48.1163	

5. CONCLUDING REMARKS

In this paper, Monte Carlo simulation (MCS) combined with a deep feedforward neural network (DFNN) is used to predict the natural frequencies of truss structures with uncertain parameters. The uncertain parameters—Young's modulus, mass density, and cross-sectional area—are modeled as random variables with uniform or normal distributions. The stochastic eigenvalue problem arising from these random parameters is sampled with MCS to obtain realizations of the natural frequencies. To reduce the high computational cost of repeatedly solving eigenvalue problems within the MCS, a DFNN surrogate predicts the eigenvalues for each sample, replacing the full eigenvalue analysis. The DFNN is trained using far fewer full-order simulations than the total number of Monte Carlo samples, enabling efficient prediction of natural frequencies. Bayesian optimization is used to tune the DFNN hyperparameters and improve its performance. The examples of truss domes, featuring different randomized parameters, demonstrate the accuracy and speed of the proposed MCS-DFNN approach compared with direct MCS. Results show that MCS-DFNN achieves comparable solution accuracy while substantially reducing computational time, with the cost savings becoming especially pronounced for large-scale structures. The MCS-DFNN framework is general and can be extended to stochastic eigenvalue analyses of other structural systems.

REFERENCES

1. Kaminski M. *The Stochastic Perturbation Method for Computational Mechanics*. Wiley; 2013.
2. Sadiku MN. *Monte Carlo Methods for Electromagnetics*. CRC Press; 2018.
3. Zakian P. Stochastic spectral cell method for structural dynamics and wave propagations. *Int J Numer Methods Eng*. 2023;**124**:4769–801.
4. Lee S. Monte Carlo simulation using support vector machine and kernel density for failure probability estimation. *Reliab Eng Syst Saf*. 2021;**209**:107481.
5. Zakian P, Khaji N, Soltani M. A Monte Carlo adapted finite element method for dislocation simulation of faults with uncertain geometry. *J Earth Syst Sci*. 2017;**126**:105.
6. Jiang S, Zhao L, Du C. Combining dynamic XFEM with machine learning for detection of multiple flaws. *Int J Numer Methods Eng*. 2021;**122**:6253–82.

[DOI: 10.22068/ijoce.2025.15.4.652]

7. Montáns FJ, Cueto E, Bathe KJ. Machine learning in computer aided engineering. In: Rabczuk T, Bathe KJ, eds. *Machine Learning in Modeling and Simulation: Methods and Applications*. Springer; 2023:1–83.
8. Tapeh ATG, Naser MZ. Artificial intelligence, machine learning, and deep learning in structural engineering: a scientometrics review of trends and best practices. *Arch Comput Methods Eng*. 2023;**30**:115–59.
9. Zakian P, Kaveh A. Multi-objective seismic design optimization of structures: a review. *Arch Comput Methods Eng*. 2024;**31**:579–94.
10. Abueidda DW, Koric S, Sobh NA. Topology optimization of 2D structures with nonlinearities using deep learning. *Comput Struct*. 2020;**237**:106283.
11. Zakian P, Niaraki Asli R. An efficient design of low-power and high-speed approximate compressor in FinFET technology. *Comput Electr Eng*. 2020;**86**:106651.
12. Schorn C, Elsken T, Vogel S, et al. Automated design of error-resilient and hardware-efficient deep neural networks. *Neural Comput Appl*. 2020;**32**:18327–45.
13. Zakian P, Niaraki Asli R. An efficient approximate multiplier: design, error analysis and application. *AEU Int J Electron Commun*. 2024;**180**:155254.
14. Ibrahim MS, Dong W, Yang Q. Machine learning driven smart electric power systems: current trends and new perspectives. *Appl Energy*. 2020;**272**:115237.
15. Gholizadeh S, Aligholizadeh V, Mohammadi M. Neural network-based reliability assessment of seismic moment frames. *Int J Optim Civ Eng*. 2014;**4**:1–26.
16. Zakian P. An efficient stochastic dynamic analysis of soil media using radial basis function artificial neural network. *Front Struct Civ Eng*. 2017;**11**:470–79.
17. Kaveh A, Khavaninzadeh N. Efficient training of two ANNs using four meta-heuristic algorithms for predicting the FRP strength. *Structures*. 2023;**52**:256–72.
18. Kaveh A, Amiri A. Efficient and interpretable seismic demands prediction for RC frames: a physics-informed, region-specific approach. *Asian J Civ Eng*. 2025.
19. Kaveh A. *Applications of Artificial Neural Networks and Machine Learning in Civil Engineering*. Springer; 2024.
20. Kim S, Shin H. Data-driven multiscale finite-element method using DNN and proper orthogonal decomposition. *Eng Comput*. 2024;**40**:661–75.
21. Mai HT, Kang J, Lee J. A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior. *Finite Elem Anal Des*. 2021;**196**:103572.
22. Mai HT, Lee S, Kim D, et al. Optimum design of nonlinear structures via deep neural network-based parameterization framework. *Eur J Mech A Solids*. 2023;**98**:104869.
23. Kamiński MM, Szafran J. Random eigenvibrations of elastic structures by the response function method and the generalized stochastic perturbation technique. *Arch Civ Mech Eng*. 2009;**9**:5–32.
24. Scheidt JV, Purkert W. *Random Eigenvalue Problems*. North Holland; 1983.
25. Ibrahim RA. Structural dynamics with parameter uncertainties. *Appl Mech Rev*. 1987;**40**:309–28.

[DOI: 10.22068/ijoce.2025.15.4.652]

26. Hollot CV, Bartlett AC. On the eigenvalues of interval matrices. In: *Proc 26th IEEE Conf Decision Control*. 1987:794–99.
27. Chen SH, Qiu Z, Liu Z. Perturbation method for computing eigenvalue bounds in structural vibration systems with interval parameters. *Commun Numer Methods Eng*. 1994;**10**:121–34.
28. Qiu Z, Wang X, Friswell MI. Eigenvalue bounds of structures with uncertain-but-bounded parameters. *J Sound Vib*. 2005;**282**:297–312.
29. Gao W. Interval natural frequency and mode shape analysis for truss structures with interval parameters. *Finite Elem Anal Des*. 2006;**42**:471–477.
30. Modares M, Mullen RL, Muhanna RL. Natural frequencies of a structure with bounded uncertainty. *J Eng Mech*. 2006;**132**:1363–71.
31. Angeli P, Barazza F, Blanchini F. Natural frequency intervals for vibrating systems with polytopic uncertainty. *J Sound Vib*. 2010;**329**:944–59.
32. Zakian P, Zakian P. Prediction of natural frequencies for truss structures with uncertainty using the support vector machine and Monte Carlo simulation. *Int J Optim Civ Eng*. 2024;**14**:211–28.
33. Qiu Z, Yang D, Elishakoff I. Probabilistic interval reliability of structural systems. *Int J Solids Struct*. 2008;**45**:2850–60.
34. Gao W, Song C, Tin-Loi F. Probabilistic interval analysis for structures with uncertainty. *Struct Saf*. 2010;**32**:191–99.
35. Wang C, Gao W, Song C, Zhang N. Stochastic interval analysis of natural frequency and mode shape of structures with uncertainties. *J Sound Vib*. 2014;**333**:2483–503.
36. Zhou X, Liu H, Shi C, Liu J. *Deep Learning on Edge Computing Devices: Design Challenges of Algorithm and Architecture*. Elsevier; 2022.
37. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016.
38. Thai HT. Machine learning for structural engineering: a state-of-the-art review. *Structures*. 2022;**38**:448–91.
39. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. *Adv Neural Inf Process Syst*. 2012;**25**.
40. Kaveh A. *Optimal Structural Analysis*. 2nd ed. Wiley; 2006.
41. Kaveh A. *Computational Structural Analysis and Finite Element Methods*. Springer; 2013.
42. Zakian P. Design optimization of moment frame structures based on natural frequency constraints using the adaptive charged system search algorithm. *Int J Optim Civ Eng*. 2021;**11**:581–97.
43. Koohestani K, Kaveh A. Efficient buckling and free vibration analysis of cyclically repeated space truss structures. *Finite Elem Anal Des*. 2010;**46**:943–48.
44. Degertekin SO, Bayar GY, Lamberti L. Parameter free Jaya algorithm for truss sizing-layout optimization under natural frequency constraints. *Comput Struct*. 2021;**245**:106461.

[DOI: 10.22068/ijoce.2025.15.4.652]

45. Kaveh A, Biabani Hamedani K, Kamalinejad M. Improved slime mould algorithm with elitist strategy and its application to structural optimization with natural frequency constraints. *Comput Struct*. 2022;**264**:106760.
46. Abbasi M, Zakian P. Optimal design of truss domes with frequency constraints using seven metaheuristic algorithms incorporating a comprehensive statistical assessment. *Mech Adv Mater Struct*. 2024;**31**:12533–59.

[DOI: 10.22068/ijoce.2025.15.4.652]