A novel data-collection strategy and automatic hyperparameter tuning of machine learning using Bayesian optimization

Hau T. Mai[a,1], Jaehong Lee[a,*]

[a]*Deep Learning Architectural Research Center, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Republic of Korea*

Abstract

Machine learning (ML) is a powerful tool in many diffenent fields. However, its training process is still challenging when applied to different types of problems. Specifically, the hyperparameters tuning and data play an important role in improving the model accuracy. In this paper, an multi-infill criterion is developed to collect the data and Bayesian optimization (BO) is utilized to tune hyperparameters of a ML model. Accordingly, the machine learning-based surrogate model replace finite element analyses (FEAs) to predict the structural responses, and then Differential evolution (DE) algorithm is used to resolve the structural optimization problems. Three numerical examples are investigated to demonstrate the efficiency of the proposed method.

Keywords: Bayesian optimization (BO), Tuning hyperparameter, Neural network, Multi-infill criterion, Structural optimization

1. Introduction

ML application to computational mechanics has special attention of researchers [1]. And, many studies have been carried out on issues relating such as structural analysis [2], fracture prediction [3, 4], nonlinear analysis [5], structural optimization [6], etc. It requires a large enough dataset to achieve good performance [3]. So, the training data depend strongly on the problem and data collection techniques, especially for expensive problems. On the other hand, hyperparameter optimization (HPO) for such models is also a computationally expensive problem and determine the efficiency of ML model [7]

To overcome this problem, decision-theoretic algorithms including Grid search (GS) and Random search (RS) are used to find hyperparameters in the search space [8], but it spends a lot of time evaluating poorly-performing areas of the design space. In order to deal with these shortcomings, BO with different surrogate models such as Gaussian process (GP), Random forest (RF), and Tree-structured Parzen Estimator (TPE) have received attentions in the HPO field. Its effectiveness in solving optimization problems for costly objective functions with fewer iterations has shown in many studies [9].

In this study, an new data-collection strategy is presented to combine HPO into the GP-relied Bayesian framework to build ML model. The ML-based surrogate model is used to estimate structural responses replacing the conventional FEAs. And, differential evolution (DE) algorithm is utilized to resolve structural optimization problems. In which, a multi-infill criterion based on BO with constraints is developed to provide maximum information about the

*Corresponding author. *E-mail: jhlee@sejong.ac.kr*

[1]*E-mail: maitienhaunx@gmail.com*

feasible region. The expected improvement (EI) is used to improve the ML model through tuning hyperparameters. Three numerical examples are investigated to demonstrate the validity of the presented method.

The remainder of this paper is organized as follows. A brief description of the ML model is presented in Section 2. Section 3 provides the BO algorithm for the HPO problem. A new multi-infill criterion is suggested in Section 4. Next, three numerical examples are investigated in Section 5. Finally, some conclusions are summarized in Section 6.

2. Machine learning

The ML model is designed to model the relationship of an input-output dataset through a training process by analyzing data and adjusting weights and biases. In this study, the neural network (NN) is utilized to predict the response mechanical responses of truss structures due to its prominent features [10]. The basic architecture of a fully connected NN comprises one input layer, one or more hidden layers, and one output layer. Each layer contains a number of neurons, and there is connection between each two neurons of two consecutive layers via weights and biases as in Fig. 1. The output value at each neuron of the ith layer is given by Eq. (1)

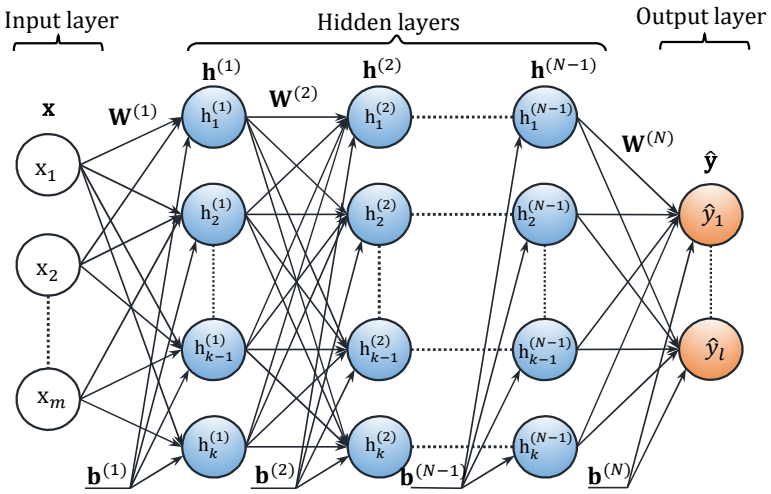

Fig. 1. Architecture of a fully connected neural network.

$$\mathbf{h}^{(i)} = \mathcal{G}\left(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}\right) \qquad \forall i = 0,\ 1,\ ...,\ N, \tag{1}$$

where $(N+1)$ is the total number of layers; $\mathbf{h}^{(0)} = \mathbf{x}$ is the input vector; $\mathbf{h}^{(N)} = \hat{\mathbf{y}}$ stands for the output vector; $\mathbf{h}^{(i)}$ symbolizes the output vector of the ith hidden layer for $i \neq 0, N$; $\mathbf{W}^{(i)}$ denotes the weight matrix connecting the $(i-1)$th layer to ith layer; $\mathbf{b}^{(i)}$ indicates the bias vector of the ith layer; $\mathcal{G}(.)$ is the activation function. It should be noted that the linear activation function of the output layer is chosen for regression problems [11].

In supervised ML, the network learns from the difference in the predicted output and the target output. To do this, an optimization algorithm is utilized to minimize the mean square error (MSE). It can be expressed as follows

$$E_{MSE} = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2, \tag{2}$$

where y_j and $\hat{y}_j$ are the target and predicted output values, respectively, and n is the number of output units multiplied by the number of training samples.

3. Hyperparameter optimization

Turning hyperparameters is computationally expensive, and BO is utilized to search optimal hyperparameters for the ML model in this work. It develops a probabilistic surrogate model to fit all currently observed samples into the target function. Next, the EI determines the next candidate points where the optimal solution is most likely to occur [7]. Accordingly, the unconstrained black-box optimization can be written as the following [12]

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^d}{\arg\min}\, f(\mathbf{x}) \tag{3}$$

where $\mathcal{X}$ is a compact set of $\mathbb{R}^d$ with d dimensions.

The distribution over functions $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}),\, k(\mathbf{x},\mathbf{x}'))$ is controlled by mean $\mu(\mathbf{x})$ and covariance $k(\mathbf{x},\mathbf{x}')$ function or *kernel*. It first put the prior on any series of sample points $\mathbf{X} = [\mathbf{x}_1, ...,\, \mathbf{x}_n]^T$ and the corresponding fitness values in the prior distribution $f(\mathbf{X}) = [f(\mathbf{x}_1)\,, ...,\, f(\mathbf{x}_n)]^T$ as follows

$$f(\mathbf{x})|\,\mathbf{x}, \mathcal{D}_n = \mathcal{N}\left(\mu_n(\mathbf{x}),\ \sigma_n^2(\mathbf{x})\right), \tag{4}$$

$$\begin{aligned} \mu_n(\mathbf{x}) &= \mu(\mathbf{x}) + \mathbf{k}^T\mathbf{K}^{-1}(\mathbf{y} - \mu(\mathbf{X})), \\ \sigma_n^2(\mathbf{x}) &= k(\mathbf{x},\mathbf{x}) - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}, \end{aligned} \tag{5}$$

where $\mu_n(.)$ and $\sigma_n^2(.)$ are the posterior mean and covariance function, respectively; $\mathbf{k} = [k(\mathbf{x},\mathbf{x}_1)\ , ...,\, k(\mathbf{x},\mathbf{x}_n)]^T$ is a vector of covariances.

In this work, the *Matérn* kernel is used in statistics and given by

$$k(\mathbf{x},\mathbf{x}') = \frac{1}{2^{(\nu-1)}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\,\|\mathbf{x}-\mathbf{x}'\|}{\theta}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}\,\|\mathbf{x}-\mathbf{x}'\|}{\theta}\right), \tag{6}$$

where $\nu \geq \frac{1}{2}$ is the shape parameter; Γ is the Gamma function; $\nu = 3/2$ is set up to construct the *Matérn* kernel function.

In this work, the EI acquisition function is utilized to address the next point at each iteration [14] as following.

$$\begin{aligned} EI(\mathbf{x}) &= \mathbb{E}\left[I(\mathbf{x})|\,\mathcal{D}_n\right], \\ &= \left(f(\mathbf{x}^*) - \mu_n(\mathbf{x})\right)\Phi\left(\tfrac{f(\mathbf{x}^*)-\mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})}\right) + \sigma_n(\mathbf{x})\,\phi\left(\tfrac{f(\mathbf{x}^*)-\mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})}\right), \end{aligned} \tag{7}$$

where $\Phi(\,)$ and $\phi(\,)$ denote the standard normal cumulative distribution function and probability density function, respectively.

To truly understand BO algorithm, the single variable test function from five initial samples is investigated. Fig. 2 shows four successive steps of the BO to obtain the near-global optimal solution. In which, the solid black line denotes the true function, the blue dashed line stands for the GP posterior mean. The respective infill function is shown by the solid red line, and the cyan area represents the 95% CI. It is easily seen that the EI value tends to increase near the minimum posterior mean, and the maximum posterior uncertainty. The new samples are determined through the combination of measured values and uncertainties.

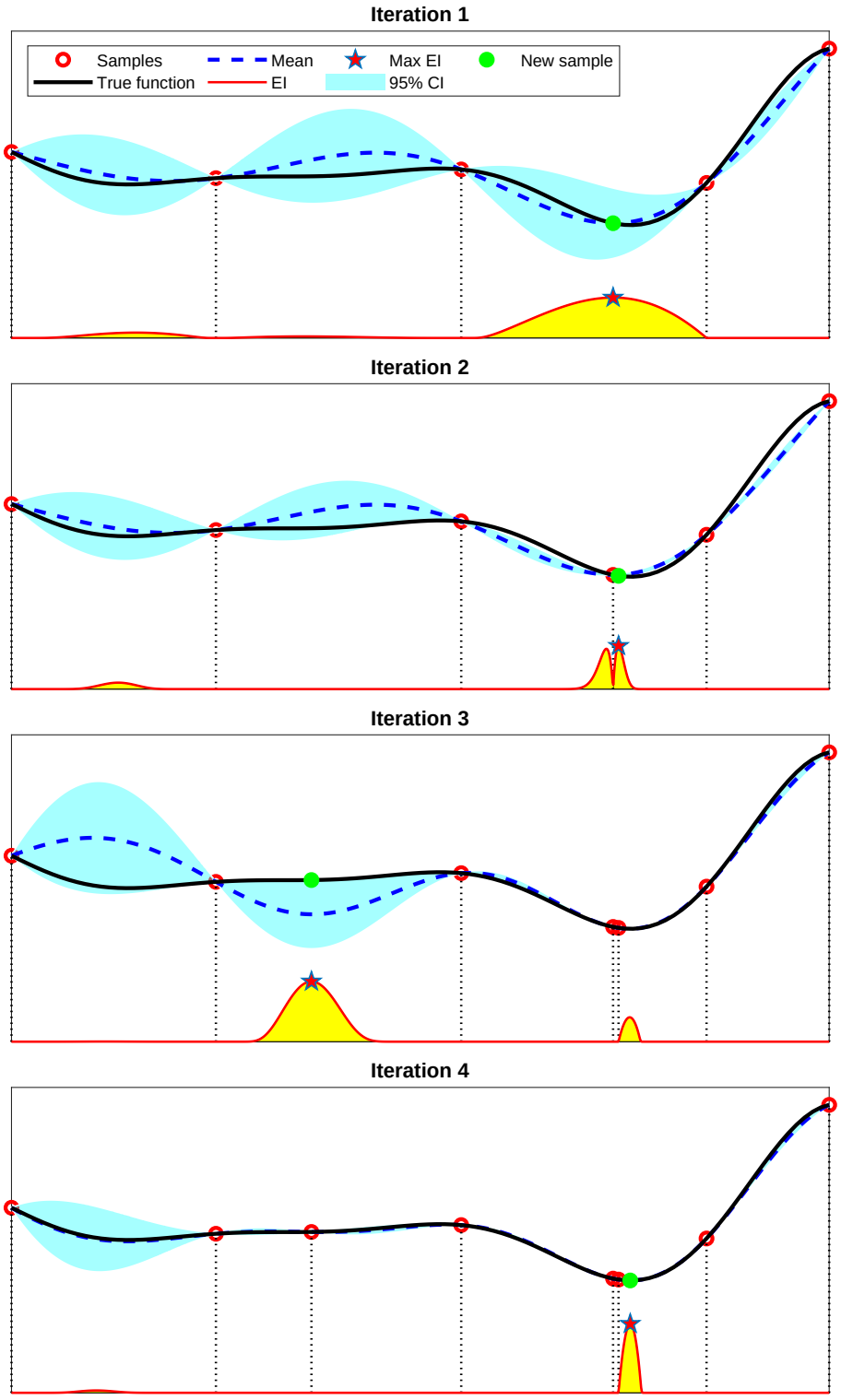

Fig. 2. The expected improvement acquisition function on a toy 1D problem.

4. Data-collection

When the data size increases can guarantee a more accurate prediction model, but the computational cost will also raise. So, a data-collection is developed based upon the maximum information obtained from data.

To achieve this objective, the generating data is divided into two step: i) searching feasible

points; ii) collecting data based on the local sampling strategy based on the feasible points. Fig. 3 illustrates the search space and the procedure of searching feasible points and promising

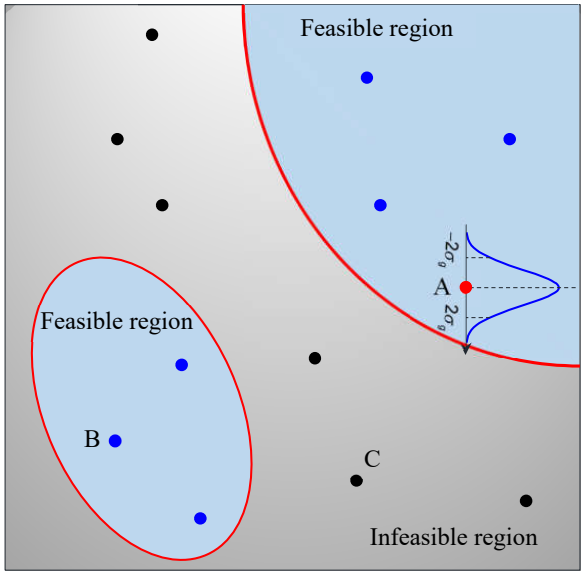

Fig. 3. Illustration of search space with feasible regions.

neighborhoods.

4.1. Searching feasible points

The constrained expected improvement (EI_C) is given by

$$EI_C(\mathbf{x}) = EI_{C_i}(\mathbf{x}) + EI_{C_r}(\mathbf{x}), \tag{8}$$

with

$$\begin{aligned}
EI_{C_i}(\mathbf{x}) &= \left(f(\mathbf{x}^*) - \mu_n(\mathbf{x})\right)\Phi\left(\frac{f(\mathbf{x}^*) - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})}\right) PF(\mathbf{x}),\\
EI_{C_r}(\mathbf{x}) &= \sigma_n(\mathbf{x})\,\phi\left(\frac{f(\mathbf{x}^*) - \mu_n(\mathbf{x})}{\sigma_n(x)}\right) PF(\mathbf{x}),\\
PF(\mathbf{x}) &= \prod_{j=1}^{m} \Pr_j\left(g_j(\mathbf{x}) \le 0\right),\\
\Pr_j\left(g_j(\mathbf{x}) \le 0\right) &= \Phi\left(\frac{-\mu_{g_j}(\mathbf{x})}{\sigma_{g_j}(\mathbf{x})}\right),
\end{aligned}$$

where $\mu_{g_j}(\mathbf{x})$ and $\sigma_{g_j}(\mathbf{x})$ are the posterior mean and covariance function of the ith constraint, respectively.

As pointed out by Haftka et al. [15], $EI_{C_i}(\mathbf{x})$ indicates point where the posterior mean of the objective function is small and low uncertainty, and the constraints are likely to be near boundaries of feasible operation. Otherwise, $EI_{C_r}(\mathbf{x})$ aims to identify point which tend to be biased towards regions that are more like to satisfy the constraints and high uncertainty of the objective function. The EI_C often fails when there is no initial feasible point in dataset.

To overcome this difficult, a multi-infill criterion is introduced in this work to achieve feasible points. This scheme helps to reduce an amount of constraint violations and to explore other feasible regions. Its formulation is written as following:

$$PFC(\mathbf{x}) \;= \sigma_n(\mathbf{x})\, PF(\mathbf{x}), \tag{9}$$

$$PIC\left(\mathbf{x}\right) \;=\; -\sum_{j=1}^{m}\max\left(0,\mu_{g_j}\left(\mathbf{x}\right)\right), \tag{10}$$

$$ISC_{14}\left(\mathbf{x}\right) \;=\prod_{j=1}^{m}\Pr_j\left(g_j\left(\mathbf{x}\right)\leq 0\right).\, D_j, \tag{11}$$

where $PFC\left(\mathbf{x}\right)$ is combined by the covariance function and the probability of feasibility function; $PIC\left(\mathbf{x}\right)$ is the penalization function corresponding to the constraint set; $ISC_{14}\left(\mathbf{x}\right)$ is the infill sampling criterion for disconnected feasible regions [16]; D is the distance to the nearest feasible point, and given by

$$D=\min_{\mathbf{x}_{feas}}\left(\frac{\left\|\mathbf{x}_{feas}-\mathbf{x}\right\|}{range}\right), \tag{12}$$

in which *range* is the lag distance at which it reaches the sill from variogram model. Interested readers are suggested to consult Refs. [16] for more details.

According to the proposed infill criterion, if the feasible point is not found in the data, $PFC\left(\mathbf{x}\right)$ and $PIC\left(\mathbf{x}\right)$ will be used as an alternative to obtain two infill sample points. In this case, providing that there exists a feasible region, two feasible sample points will be identified. More specifically, the first point is located in the sparsely sampled area by maximizing $PFC\left(\mathbf{x}\right)$, so $\sigma_n\left(\mathbf{x}\right)$ characterizes the sample density of the objective function in the design space. And the other point is indicated the corresponding minimum value of the total constraints $PIC\left(\mathbf{x}\right)$. When the models fit poorly and do not have a feasible region, $PIC\left(\mathbf{x}\right)$ is utilized to improve at locations violated constraints. On the contrary, once an initial feasible point is found, Eq. (9) and Eq. (11) are employed to determine infill sample points. $ISC_{14}\left(\mathbf{x}\right)$ aims to effectively explore other feasible regions.

4.2. Local sample points

the local sample points based on the feasible sample points and the distance criterion to collect data. To achieve this objective, the feasible sample points is the center of local domains, and then LHS technique is used to create several subsamples in the local domains. The boundary values $[\mathbf{x}_{lb}^{i};\ \mathbf{x}_{ub}^{i}]$ of each subdomain are expressed below

$$\left[\mathbf{x}_{lb}^{i};\ \mathbf{x}_{ub}^{i}\right]=\left[\mathbf{x}_{fes}^{i}-\frac{dis}{2};\ \mathbf{x}_{fes}^{i}+\frac{dis}{2}\right]\cap\left[\mathbf{x}_{lb};\ \mathbf{x}_{ub}\right], \tag{13}$$

in which

$$\begin{cases} dis=\left|\mathbf{x}_{fes}^{i}-\mathbf{x}_{fes}^{close}\right| & if\ \ \left\|x_{fes}^{i}-\mathbf{x}_{fes}^{close}\right\|>\left\|\xi\left(\mathbf{x}_{ub}-\mathbf{x}_{lb}\right)\right\|,\\ dis=\xi\left(\mathbf{x}_{ub}-\mathbf{x}_{lb}\right) & \text{otherwise},\end{cases}$$
$$\min_{\mathbf{x}_{fes}^{close}}\left(\left\|\left(\mathbf{x}_{fes}^{i}-\mathbf{x}_{fes}^{j}\right)\right\|\right)\ \ j=1,\,2,\,...,i-1,i+1,...,\,n_{fes},$$

where ξ is a coefficient chosen as 0.05 in this work, $\mathbf{x}_{fes}^{close}$ is the closest feasible point to $\mathbf{x}_{fes}^{i}$, n_{fes} is the number of feasible points.

5. Results and discussion

Tuning hyperparameters is shown in Table 2. The data is divided into three subsets, i.e. 60% for training, 20% for validation, and 20% for testing. the two different datasets are considered in this study. The first data denoted by DS1 has 4000 samples, which is created by the LHS technique. The other data denoted by DS2 is created by the proposed approach.

Table 1
Data for optimization problems of truss structures.

Problem	Modulus of elasticity E (N/m^2)	Material density ρ (kg/m^3)	Cross-sectional area bounds (cm^2)	Frequency constraints (Hz)
10-bar planar truss	6.98×10^{10}	2770	$0.645 \leq A \leq 50$	$7 \leq f_1$; $15 \leq f_2$; $20 \leq f_3$
72-bar space truss	6.98×10^{10}	2770	$0.645 \leq A \leq 20$	$f_1 = 4$; $6 \leq f_3$

Table 2
Specifics of the configuration space for hyperparameters.

Hyperparameter	Search space		Type
	RS, BOHB, BO-TPE, BO-GP	GS	
No. of hidden layers	[1, 9]	[1, 5]	Integer
No. of nodes in hidden layers	[10, 300]	[20, 50, ..., 290]	Integer
Batch size	[25, 26, ..., 210]	64	Integer
Learning rate	$[10^{-7}, 10^{-2}]$	0.01	Real
Learning rate decay	$[10^{-9}, 10^{-5}]$	10^{-6}	Real
Activation function	[Tanh, Softplus, ReLU, Sigmoid]	[ReLU, Sigmoid]	Categorical

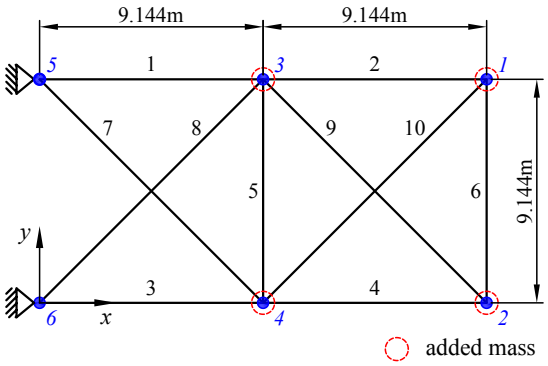

Fig. 4. Schematic of the planar 10-bar truss structure.

5.1. 10-bar planar truss

The first problem is 10-bar planar truss as depicted in Fig. 4. The cross-sectional areas of all bars are assumed to be continuous design variables. A non-structural mass of 454 kg is placed on each free node of the structure. Table 1 shows material properties, frequency constraints, and cross-sectional area bounds. It has been previously examined by many authors [17]. The developed paradigm finds 121 feasible sampling points and 2720 sample points in total for the DS2 dataset. Fig. 5 shows the data distribution and the correlation of structural frequencies. The results of the optimal hyperparameters obtained by combining the HPO method with two data are reported in Table 3 and Fig. 6. The MSE values obtained from the DS1 are always larger than the DS2 when using the same HPO method. Otherwise, the values of the BO-GP is lower than those produced by the remaining methods. Table 5 reported the cross-validation errors evaluated via MSE and RMSE for statistical outcomes including mean, standard deviation, and 95% CI. The BO-GP-DS2 is the most effective model because it yields the smallest cross-validation error.

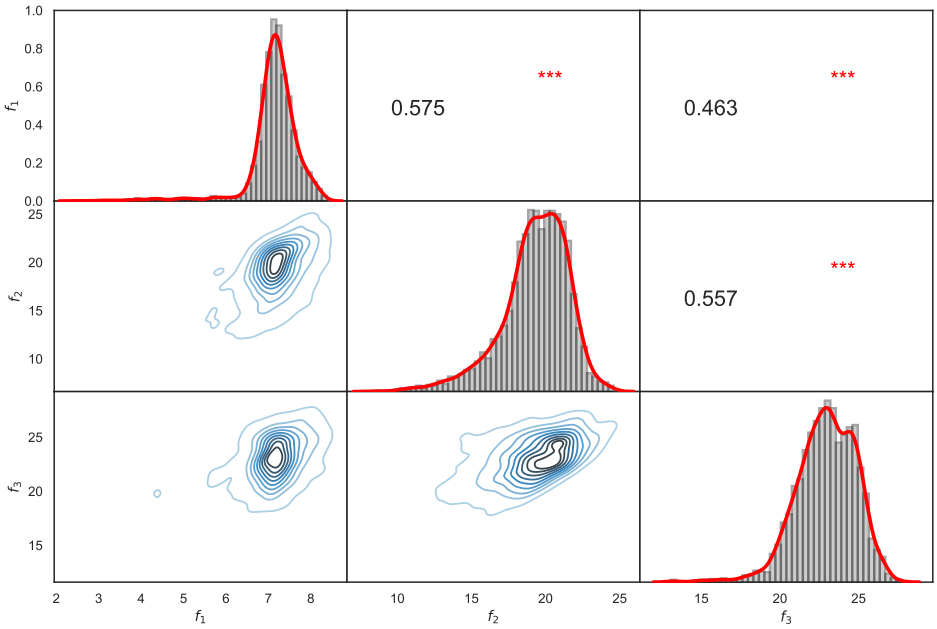

Fig. 5. Data distribution and correlation between natural frequencies of the 10-bar planar truss structure.

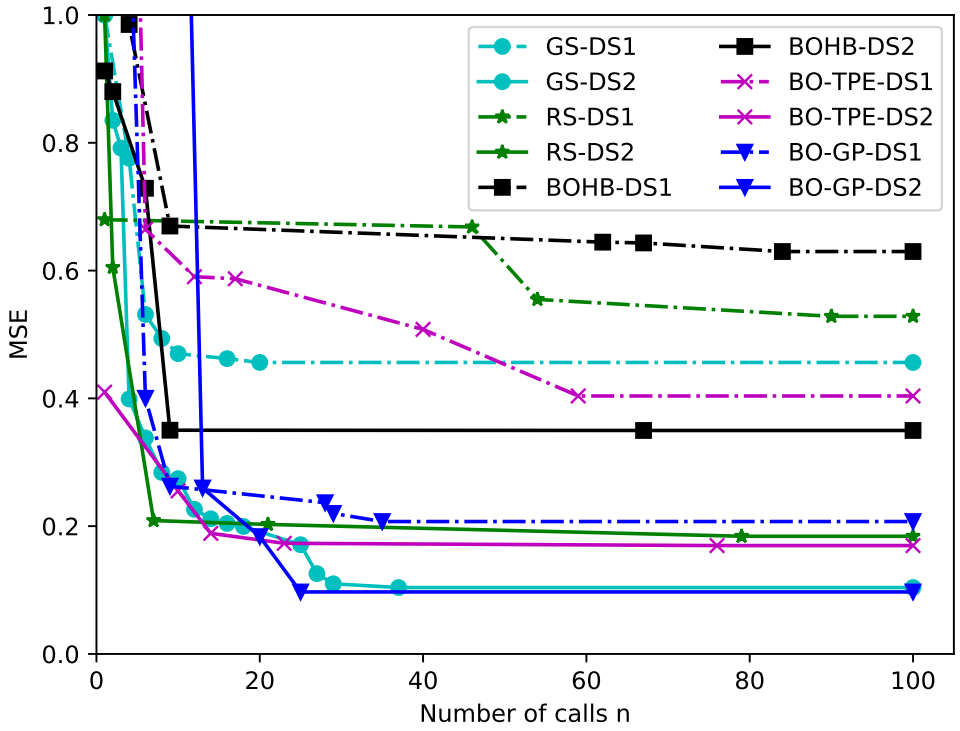

Fig. 6. The convergence rates of the hyperparameter algorithms for the 10-bar planar truss.

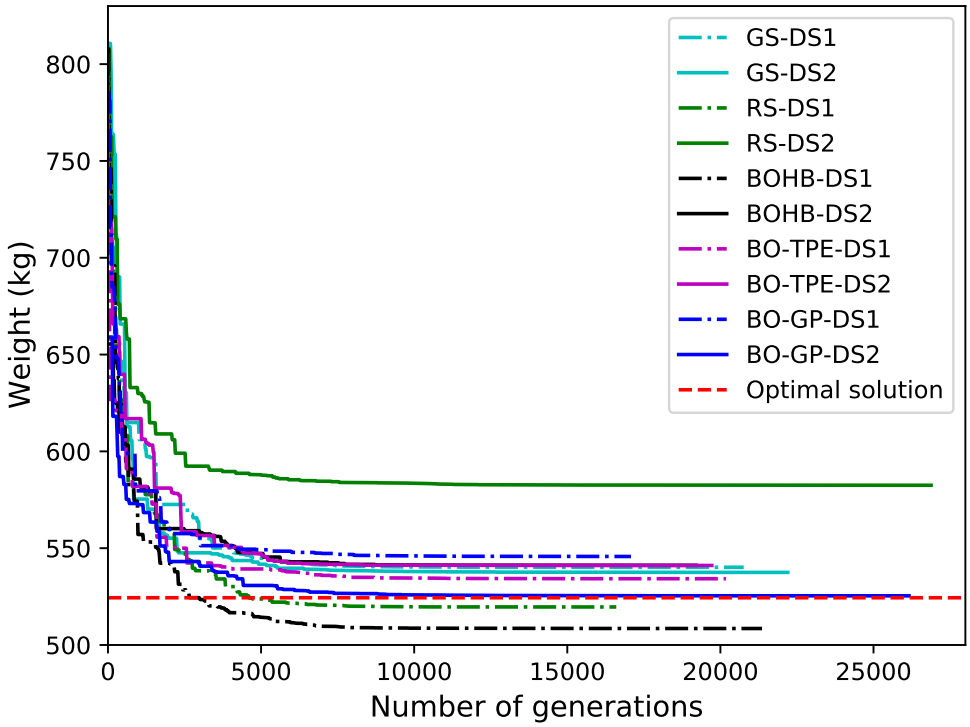

Fig. 7. The weight convergence histories of the 10-bar planar truss obtained using the DE.

Table 4 presents a comparison of optimal results obtained by using different ML models. In which, the gained cross-sectional areas are used to achieve the first three natural frequencies by FEM. The optimization results reveal that the DS2 dataset generally outperforms while the number of samples reaches two-third of the DS1 data. It demonstrates the effectiveness of the suggested method. It is noticed that in this case, the results obtained from the BO-GP are better than others, and this can be easily seen through the violated frequencies. The result attained from the BO-GP-DS2 is of the smallest optimal weight compared with the others. The constraint values agree well with the FEM, and close to the constraint limit. Fig. 7 shows the weight convergence histories obtained using different combinations for this problem. Clearly, the convergence rate of the BO-GP-DS2 shows a good agreement with the optimal solution obtained by FEM.

5.2. 72-bar space truss

Next, 72-bar space truss as depicted in Fig. 8 is considered for the next problem. Non-structural masses of 2270kg are applied at the upper nodes of the structure. By using the proposed approach, 137 feasible sampling points and 3040 samples in the DS2 data are found. The output data distribution is highlighted in Fig. 9,

The optimal hyperparameters and the convergence history of various algorithms are reported in Table 6 and Fig. 10, respectively. BO-GP-DS2 yields the smallest MSE with the least number of iterations.

The comparison of the obtained results achieved by the proposed method and other methods is presented in Table 7. It is easily seen that the BO-GP-DS2 yields the solution closest to the FEM solution with the relative error of the weight being small 0.28%.

Table 3
Optimal hyperparameters of ANN obtained by different methods for the 10-bar planar truss.

Hyperparameters	GS		RS		BOHB		BO-TPE		BO-GP	
	DS1	DS2	DS1	DS2	DS1	DS2	DS1	DS2	DS1	DS2
No. of hidden layers	2	1	3	2	1	1	5	3	1	2
No. of nodes in hidden layers	260	290	223	258	202	210	247	280	300	300
Batch size	64	64	64	256	64	64	64	64	0.01	32
Learning rate	0.01	0.01	4.83E-04	2.90E-03	0.01	1.37E-03	1.02E-03	8.70E-03	1.00E-02	1.00E-02
Learning rate decay	1.00E-06	1.00E-06	7.84E-08	4.28E-09	1.00E-06	2.30E-07	7.77E-06	2.80E-07	1.00E-05	2.33E-09
Activation function	Sigmoid	ReLU	Tanh	Tanh	Sigmoid	Sigmoid	ReLU	ReLU	Softplus	Sigmoid

Table 4
Comparison of optimal results with different hyperparameter tuning methods for the 10-bar truss.

Design variable	GS		RS		BOHB		BO-TPE		BO-GP		
A_i(cm^2)	DS1	DS2	DS1	DS2	DS1	DS2	DS1	DS2	DS1	DS2	FEM
1	18.25	40.00	28.78	37.73	29.07	39.85	36.41	39.98	40.00	30.19	35.17
2	39.84	11.28	24.60	10.45	24.41	13.08	19.38	12.97	12.76	15.85	14.70
3	39.99	40.00	35.67	31.52	39.99	40.00	34.39	40.00	40.00	39.14	35.10
4	34.28	12.62	21.07	16.75	18.23	13.77	15.24	13.78	13.85	15.83	14.69
5	0.65	3.43	0.65	4.90	0.65	0.65	11.01	0.65	4.54	0.65	0.65
6	0.65	6.56	2.12	5.38	0.65	6.62	10.16	6.77	5.25	5.31	4.56
7	39.99	28.90	24.08	25.77	35.69	31.53	17.92	31.54	19.29	24.54	23.73
8	0.65	20.07	31.76	30.37	25.06	21.16	20.18	20.97	32.13	21.64	23.67
9	14.93	10.85	2.88	20.59	0.65	9.16	15.97	9.26	11.21	11.42	12.39
10	0.69	9.70	6.49	10.42	0.65	8.64	5.55	8.58	7.41	13.44	12.42
Weight(kg)	540.14	537.49	519.63	582.50	508.51	541.15	534.22	541.07	545.72	525.39	524.40
f_1(Hz)	3.3426	6.9465	5.8788	7.1813	2.9112	6.8916	6.6008	6.8885	6.9295	6.9486	6.9990
f_2(Hz)	7.0663	18.1698	14.3660	18.0421	7.8575	16.6023	18.3988	16.5888	18.9451	16.2802	16.1897
f_3(Hz)	14.7137	19.1679	16.8857	20.6438	15.4854	19.8400	20.7978	19.7995	19.5686	20.0006	19.9946

Table 5
Results of MSE and RMSE for the 10-bar truss.

	Data	Statistics	Training		Validation	
			MSE	RMSE	MSE	RMSE
GS	DS1	Mean	0.1390	0.3717	0.4872	0.6976
		Std	0.0215	0.0292	0.0355	0.0253
		95% CI	0.124-0.154	0.351-0.393	0.462-0.513	0.68-0.716
	DS2	Mean	0.1256	0.3538	0.1752	0.4171
		Std	0.0146	0.0205	0.0311	0.0364
		95% CI	0.115-0.136	0.339-0.369	0.153-0.197	0.391-0.443
RS	DS1	Mean	0.1836	0.4277	0.5178	0.7194
		Std	0.0220	0.0261	0.0266	0.0184
		95% CI	0.168-0.199	0.409-0.446	0.499-0.537	0.706-0.732
	DS2	Mean	0.2026	0.4499	0.3044	0.5504
		Std	0.0124	0.0137	0.0454	0.0393
		95% CI	0.194-0.211	0.44-0.46	0.272-0.337	0.522-0.579
BOHB	DS1	Mean	0.2243	0.4723	0.6493	0.8056
		Std	0.0343	0.0371	0.0329	0.0204
		95% CI	0.2-0.249	0.446-0.499	0.626-0.673	0.791-0.82
	DS2	Mean	0.3536	0.5942	0.4560	0.6746
		Std	0.0298	0.0243	0.0435	0.0322
		95% CI	0.332-0.375	0.577-0.612	0.425-0.487	0.652-0.698
BO-TPE	DS1	Mean	0.1364	0.3685	0.4318	0.6565
		Std	0.0199	0.0268	0.0398	0.0307
		95% CI	0.122-0.151	0.349-0.388	0.403-0.46	0.634-0.678
	DS2	Mean	0.1952	0.4416	0.2659	0.5149
		Std	0.0126	0.0143	0.0296	0.0286
		95% CI	0.186-0.204	0.431-0.452	0.245-0.287	0.494-0.535
BO-GP	DS1	Mean	0.2147	0.4633	0.2964	0.5438
		Std	0.0058	0.0062	0.0296	0.0269
		95% CI	0.211-0.219	0.459-0.468	0.275-0.318	0.525-0.563
	DS2	Mean	0.1055	0.3239	0.1245	0.3506
		Std	0.0161	0.0252	0.0300	0.0421
		95% CI	0.094-0.12	0.31-0.34	0.103-0.146	0.32-0.38

Table 6
Optimal hyperparameters of ANN from each method for the 72-bar space truss.

Hyperparameters	GS	RS	BOHB	BO-TPE	BO-GP	
	DS2				DS1	DS2
No. of hidden layers	2	7	1	2	2	1
No. of nodes in hidden layers	260	147	203	225	300	300
Batch size	64	32	64	128	128	32
Learning rate	0.01	1.62E-03	1.56E-03	6.70E-04	2.17E-04	1.99E-04
Learning rate decay	1.00E-06	1.12E-08	1.48E-07	1.53E-08	1.01E-08	1.00E-09
Activation function	Sigmoid	Sigmoid	Sigmoid	ReLU	ReLU	ReLU

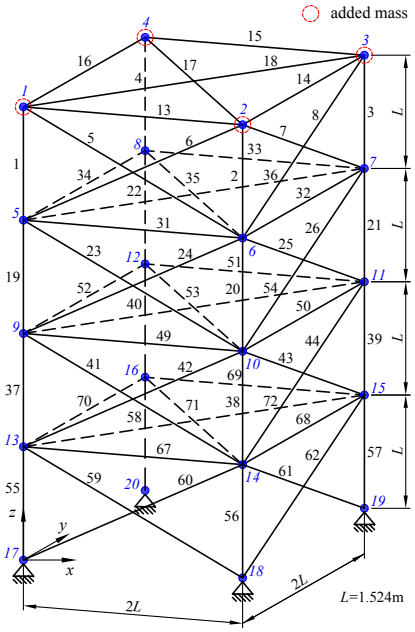

Fig. 8. The model of 72-bar space truss.

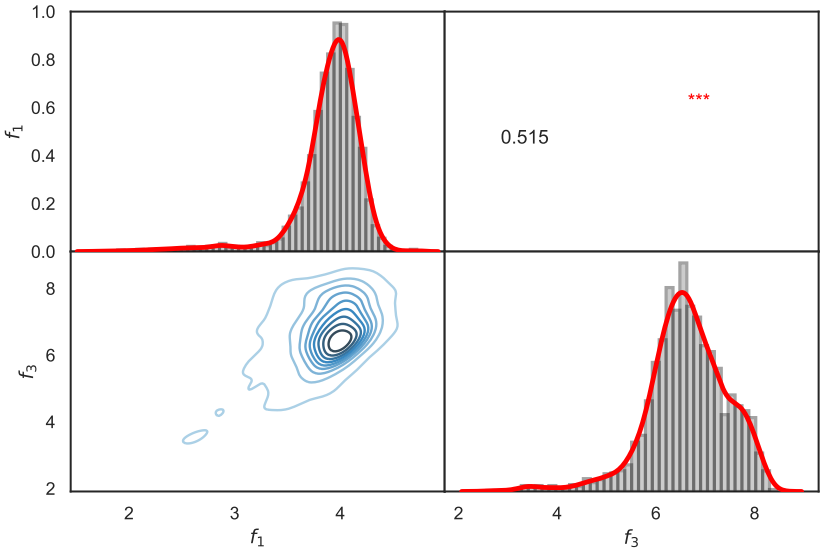

Fig. 9. Data distribution and correlation between the natural frequencies of the 72-bar space truss structure.

5.3. 52-bar dome truss

In the last example, design optimization of dome truss with geometric nonlinearity is considered in dealing with computationally expensive problems. The 52-bar dome truss has been previously examined by Saka [18] described in Fig. 11. The Young's modulus and material density are $E = 21000\text{kN/cm}^2$ and $\rho = 7.85\text{N/cm}^3$, respectively. The external force 150kN in the z-axis direction subject to nodes 6-13. The vertical displacements of all free nodes are restricted to 10mm.

Table 7
Comparison of optimal results with different hyperparameter tuning methods for the 72-bar space truss.

Design variable	GS	RS	BOHB	BO-TPE	BO-GP		FEM
A_i(cm^2)	DS2				DS1	DS2	
1	4.406	0.645	0.648	3.529	0.645	3.132	3.861
2	7.623	7.705	9.336	8.331	6.869	7.815	7.828
3	0.645	0.645	0.645	0.645	0.645	0.645	0.704
4	0.645	0.645	0.645	0.645	0.645	0.645	0.647
5	8.527	7.714	8.388	6.238	8.853	10.218	7.281
6	8.569	8.473	8.436	8.384	8.812	7.431	8.159
7	0.645	0.645	0.645	0.645	0.645	0.645	0.658
8	0.645	0.645	0.645	0.645	0.645	0.645	0.645
9	12.400	11.506	14.293	10.498	17.669	13.886	13.602
10	7.890	8.381	9.241	7.669	7.829	8.929	8.136
11	0.645	0.645	0.645	0.645	0.645	0.645	0.663
12	0.645	0.645	0.645	0.645	0.645	0.645	0.722
13	16.694	19.847	20.000	20.000	19.967	15.136	16.528
14	8.739	8.697	7.535	7.208	9.158	7.452	7.668
15	0.645	0.645	0.645	0.645	0.645	0.645	0.674
16	0.645	0.645	0.645	0.645	0.645	0.645	0.658
Weight(kg)	333.695	333.071	348.933	321.432	341.168	325.257	324.340
f_1(Hz)	4.042	3.609	3.701	3.928	3.725	4.001	3.997
f_3(Hz)	6.091	5.113	5.138	5.972	5.150	5.972	6.001

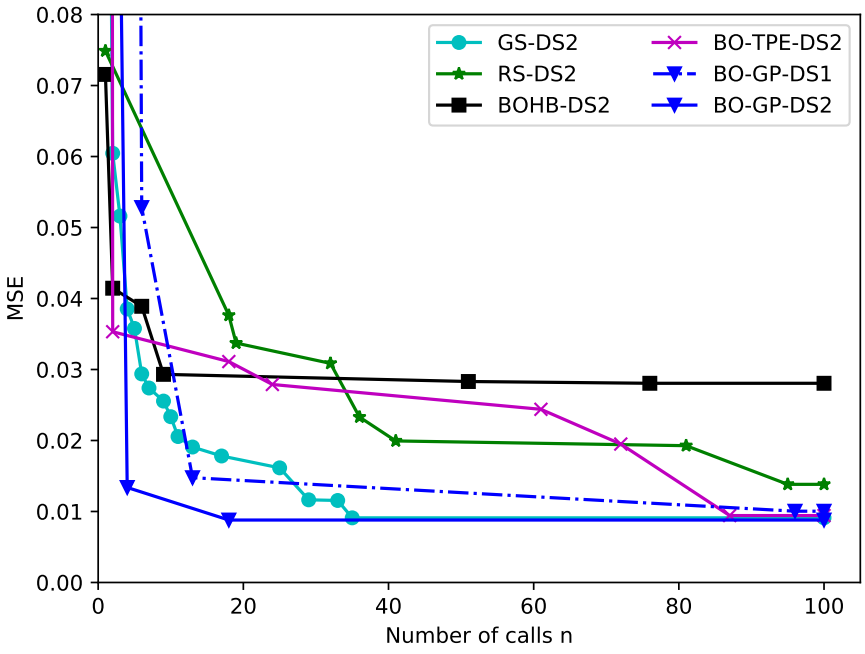

Fig. 10. The convergence rates of the hyperparameter tuning algorithms for the 72-bar space truss.

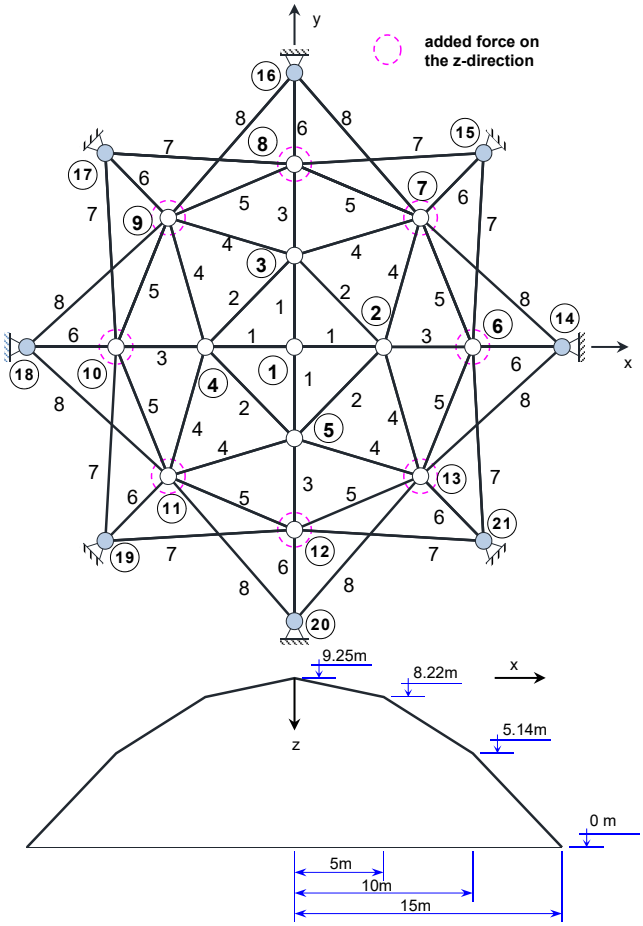

Fig. 11. The 52-bar dome truss.

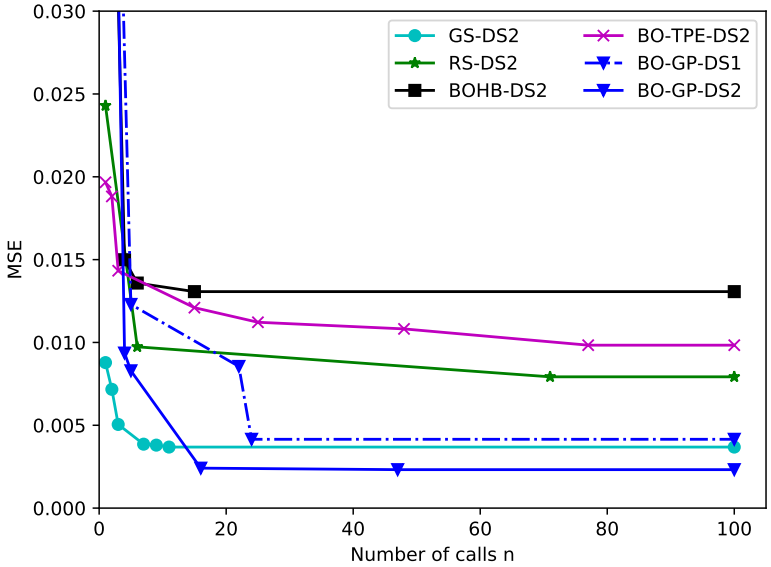

Fig. 12. The convergence rates of the hyperparameter algorithms for the 52-bar dome truss.

The suggested data-collection algorithm finds 71 feasible samples and 1570 samples for the DS2 dataset. The optimal hyperparameters and the convergence rates of various algorithms are

provided in Table 8 and Fig. 12, respectively. It also indicates that the BO-GP-DS2 model gives the smallest error.

Table 8
Optimal hyperparameters of ANN from each method for the dome structure with 52 members.

Hyperparameter	GS	RS	BOHB	BO-TPE	BO-GP	
	DS2				DS1	DS2
No. of hidden layers	1	2	2	3	1	1
No. of nodes in hidden layers	170	38	204	218	300	151
Batch size	64	64	64	128	32	32
Learning rate	0.01	1.62E-03	2.17E-03	5.23E-03	0.01	5.40E-03
Learning rate decay	1.00E-06	1.00E-05	9.23E-08	2.89E-09	4.04E-07	1.00E-05
Activation function	Sigmoid	ReLU	ReLU	Softplus	Softplus	ReLU

As the previously indicated example, Table 10 summarizes the optimal results obtained by some of the above networks for comparison. First, as can be seen from Table 10, the constraints obtained on DS2 are not violated, whereas the fourth constraint in DS1 is violated (19.334mm $\geq$ 10mm). The optimal solution from BO-GP-DS2 is very close to FEM with a small prediction error.

The weight convergence histories obtained using different models for this example are given in Fig. 13. The BO-GP-DS2 shows its accuracy and effectiveness in significantly remarkably the computational cost.

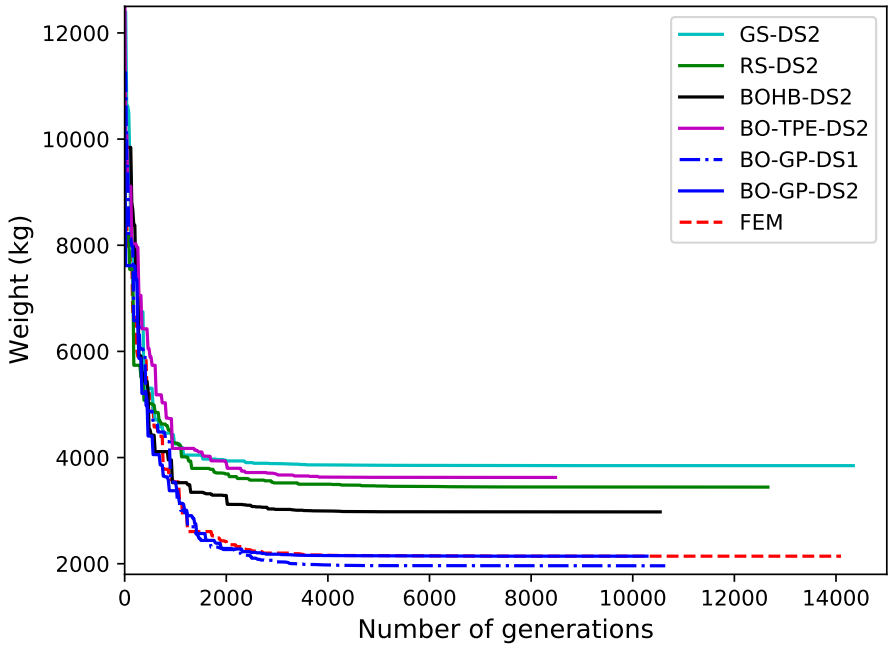

Fig. 13. The weight convergence histories of the 120-bar dome truss obtained using the DE.

Table 9
Comparison of computational time for the dome structure with 52 members.

Times(s)	GS	RS	BOHB	BO-TPE	BO-GP		FEM
	DS2				DS1	DS2	
Data generation	3194	3194	3194	3194	5067	3194	68408
Training	25916	34264	16668	29880	26512	27488	
Optimization	19	17	15	12	20	14	
Total	29129	37475	19877	33086	31599	30696	68408

Table 10
Comparison of optimal results with different hyperparameter tuning methods for the dome structure with 52 members.

Design variable	GS	RS	BOHB	BO-TPE	BO-GP		FEM	Saka [18]
A_i(cm^2)	DS2				DS1	DS2		
1	22.086	2.000	2.000	2.000	2.000	2	2.000	81.820
2	2.001	2.000	2.000	2.000	2.000	2	2.000	22.410
3	21.049	21.536	2.000	2.000	2.000	2	2.000	33.580
4	2.000	2.000	2.000	2.000	2.000	2	2.000	14.450
5	28.933	23.369	25.183	35.250	12.083	16.753	16.672	10.640
6	28.277	32.197	27.886	31.560	10.460	17.869	17.585	25.160
7	2.000	2.000	2.000	2.000	7.385	2.3013	2.519	2.000
8	2.000	2.000	2.000	2.000	2.000	2	2.000	2.000
Weight(kg)	3847.422	3443.526	2976.287	3625.623	1961.140	2142.41	2141.870	5161
w_1(mm)	1.104	0.171	0.646	1.046	3.907	1.206	1.326	-2.772
w_2(mm)	0.923	0.056	0.466	0.915	0.826	0.742	0.719	-2.826
w_6(mm)	6.798	7.206	7.252	5.815	7.735	10	10	**13.045**
w_7(mm)	5.723	5.866	6.182	5.039	**19.334**	9.648	10	9.491

6. Conclusions

In this work, an new multi-infill criterion has been successfully developed for handling constraints and providing maximum information about feasible samples. The ML model with optimized tuning hyperparameters is utilized to optimize truss structures with linear and non-linear responses. The attained results indicated that the combination of BO with data-collection strategy is reliable enough to handle structural optimization problems.

Acknowledgment

This research was supported by a grant (NRF- 2020R1A4A2002855) from NRF (National Research Foundation of Korea) funded by MEST (Ministry of Education and Science Technology) of Korean government.

References

[1] J. Alam, L. Berke, Functional approximation using artificial neural networks in structural mechanics, NASA STI/Recon Technical Report N 94 (1993) 11254.

[2] 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.

[3] S. Feng, H. Zhou, H. Dong, Using deep neural network with small dataset to predict material defects, Materials & Design 162 (2019) 300–310.

[4] A. T. Huynh, Q. D. Nguyen, Q. L. Xuan, B. Magee, T. Chung, K. T. Tran, K. T. Nguyen, A machine learning-assisted numerical predictor for compressive strength of geopolymer concrete based on experimental data and sensitivity analysis, Applied Sciences 10 (2020) 7726.

[5] E. Samaniego, C. Anitescu, S. Goswami, V. M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, Computer Methods in Applied Mechanics and Engineering 362 (2020) 112790.

[6] S. Lee, H. Kim, Q. X. Lieu, J. Lee, Cnn-based image recognition for topology optimization, Knowledge-Based Systems (2020) 105887.

[7] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing 415 (2020) 295–316.

[8] T. Yu, H. Zhu, Hyper-parameter optimization: A review of algorithms and applications, arXiv preprint arXiv:2003.05689 (2020).

[9] S. Shin, Y. Lee, M. Kim, J. Park, S. Lee, K. Min, Deep neural network model with bayesian hyperparameter optimization for prediction of nox at transient conditions in a diesel engine, Engineering Applications of Artificial Intelligence 94 (2020) 103761.

[10] S. Lee, J. Ha, M. Zokhirova, H. Moon, J. Lee, Background information of deep learning for structural engineering, Archives of Computational Methods in Engineering 25 (2018) 121–129.

[11] M. Rafiq, G. Bugmann, D. Easterbrook, Neural network design for engineering applications, Computers & Structures 79 (2001) 1541–1552.

[12] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, J. P. Cunningham, Bayesian optimization with inequality constraints, in: ICML, volume 2014, 2014, pp. 937–945.

[13] A. Fradi, C. Samir, Bayesian cluster analysis for registration and clustering homogeneous subgroups in multidimensional functional data, Communications in Statistics-Theory and Methods (2020) 1–17.

[14] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (1998) 455–492.

[15] R. T. Haftka, D. Villanueva, A. Chaudhuri, Parallel surrogate-assisted global optimization with expensive functions–a survey, Structural and Multidisciplinary Optimization 54 (2016) 3–13.

[16] M. Sasena, P. Papalambros, P. Goovaerts, Global optimization of problems with disconnected feasible regions via surrogate modeling, in: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2002, p. 5573.

[17] Q. X. Lieu, D. T. Do, J. Lee, An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints, Computers & Structures 195 (2018) 99–112.

[18] M. Saka, M. Ulker, Optimum design of geometrically nonlinear space trusses, Computers & structures 42 (1992) 289–299.