# Efficient framework for structural reliability analysis based on adaptive ensemble learning paired with subset simulation

Truong-Thang Nguyen[a,1], Viet-Hung Dang[a,*] and Huan Xuan Nguyen[b,2]

[a]Faculty of Building and Industrial Construction, Hanoi University of Civil Engineering, Vietnam
[b]London Digital Twin Research Centre, Faculty of Science and Technology, Middlesex University, London, UK

ARTICLE INFO

*Keywords*:
Reliability analysis
machine learning
statistic
structural analysis
numerical simulation.

ABSTRACT

This study develops a novel, accurate and efficient framework for the reliability analysis of multi-component civil structures, which involves many random variables and repeated time-consuming structural analysis processes. The simple yet effective idea is to construct a binary classification surrogate model to detect the structure's condition (failure/safe) given structural parameters, external loads, and predefined safety thresholds. However, building a surrogate model that can accurately detect the structure condition is challenging because of the heavy imbalance nature of the considered problem, i.e., only a very small portion of the data corresponds to the failure condition. To overcome this problem, an ensemble learning model which stacks six different classification machine/deep learning models is engineered. The ensemble model can improve the classification performance by leveraging the model diversity rather than manually tuning a set of hyperparameters. Besides, a subset simulation scheme is leveraged to address the scarcity of relevant samples, i.e., providing more training samples from the potential failure region. In addition, the sampling weighting technique is adopted to assign higher weights for samples corresponding to the class with smaller probability, allowing the training process to achieve a faster convergence rate and higher final performance. The applicability and efficiency of the proposed approach are successfully demonstrated through three case studies with different complexity and dimensionality, showing that the proposed approach can provide accurate reliability results with up to two orders of magnitudes less computational costs.

## 1. Introduction

Civil structures are expensive multi-component systems involving a large number of random variables related to external loads, material's mechanical properties, and geometric properties; therefore, it is important yet challenging to ensure their high reliability and long lifespan. Monte Carlo (MC) simulation is a de-factor method to numerically analyze the structural reliability since it does not require any simplifying assumption on the structure's behavior or the failure domain. However, the MC method suffers from two major drawbacks: i) repeatedly performing a large number of calculations and ii) the requirement of time-consuming numerical simulations such as the Finite Element Method (FEM). In order to address these two problems, there are two respective research directions plus a hybrid strategy of the two.

The first research direction is to employ a sampling strategy to sample more relevant data from failure domains based on knowledge progressively achieved during calculation iterations. Some well-known sampling strategies are the line sampling [1] which is able to identify the important direction in the input parameter space; the importance sampling [2], which introduces importance sampling density functions centered around the most probable failure points; and the subset simulation (SS) computing the final failure probability as a product of conditional intermediate failure probabilities. The pioneering work that applied SS to the structural reliability analysis was proposed by Au and Beck [3], proving that SS could estimate a very small failure probability with acceptable accuracy and with impressive reduced time complexity. After that, several authors have developed different SS variants to extend their applications and improve their performance. For example, Hsu and Ching [4] designed a parallel SS approach for addressing multiple limit states simultaneously. Alvarez et al. [5] proposed to combine SS with the random set theory to estimate the lower and upper bounds of the failure probability of structures subjected to aleatoric and epistemic uncertain inputs. Ebenuwa and Tee [6] analyzed the reliability of structures with the presence of deterministic, random, and fuzzy variables through a fuzzy-based optimized SS approach. Abdollahi et al. [7] suggested incorporating the control variate technique with SS, forming the subset control variate (SCV) method. Several examples were performed to demonstrate improved reliability results achieved by SCV. For structures exhibiting inelastic behaviors, Katafygiotis and Cheung [8] developed a two-stage SS variant where SS was only activated for samples from the inelastic region, making the two-stage SS method more efficient than the standard one.

The second research direction is to conceive a fast surrogate model in place of FEM in estimating the limit state function. A practical and easy-to-implement method is the Response Surface Modeling [9] which constructs a predictor using a polynomial function of random variables plus a random error. However, the Response Surface Modeling's prediction accuracy for a nonlinear limit state function of complex problems is limited. Another widely used surrogate model is the Kriging model, a.k.a, Gaussian Process [10,

*Corresponding author
thangnt2@huce.edu.vn (T. Nguyen); hungdv@huce.edu.vn (V. Dang); H.Nguyen@mdx.ac.uk ( Huan Xuan Nguyen)
ORCID(s): 0000-0003-3384-6427 (V. Dang)

**Table 1**
Comparison between the proposed a-eSS framework with other active learning methods in the structural reliability literature

| N$^o$ | Authors | Surrogate model | Sampling strategy | Structural application | Number of variables | Year |
|---|---|---|---|---|---|---|
| 1 | Echard et al. [18] | Kriging model | Monte Carlo | 1-dof oscillator | 6 | 2011 |
| 2 | Chojaczyk et al. [19] | ANN | Important Sampling | 3D stiffer plate | 13 | 2015 |
| 3 | Vazirizade et al. [20] | ANN | Monte Carlo | 2D story frame | 3 | 2017 |
| 4 | Ghosh et al. [21] | SVM | Monte Carlo | 2D bridge piers | 5 | 2018 |
| 5 | Hariri-Ardebili [22] | KNN/Naive Bayes | Important Sampling | Dam structure | 18 | 2018 |
| 6 | Zhang et al. [23] | Kriging model | Subset simulation | 2D truss structure | 10 | 2019 |
| 7 | Gomes and Jose [24] | ANN | Important Sampling | 1-dof oscillator | 6 | 2019 |
| 8 | Zhao et al. [25] | RBF-GA | Monte Carlo | 2D 6-story frame | 67 | 2019 |
| 9 | Mendoza Lugo et al. [26] | Bayesian Network | Monte Carlo | 3D bridge columns | 17 | 2019 |
| 10 | Roy et al. [27] | SVM | Sequential sampling | 3D 6-story building | 39 | 2020 |
| 11 | Lieu et al. [15] | Deep NN | Monte Carlo | 3D 120-bar dome | 11 | 2022 |
| 12 | Zhou and Peng [17] | DNN + Gaussian Process | Probability density evolution | 2D 10-story frame structure | 110 | 2022 |
| 13 | a-eSS (this study) | Ensemble model | Subset simulation | 3D 9-story building | 42 | 2022 |

11], which is basically a statistic-based interpolation technique leveraging the spatial correlation between data samples to interpolate unknown values. However, the computation time of the standard Kriging model increases quickly with the number of variables [12] because it requires performing multiple matrix inversion and multiplication operations. Lately, using machine learning/deep learning algorithms to build a surrogate model has gained considerable attention thanks to its high performance and efficiency. Dang et al. [13] developed a surrogate model based on the Bayesian neural network to analyze the time-varying reliability of structures. Later, the authors [14] leveraged the deep learning algorithm Long Short Term Memory to approximate the structure's time-varying responses when performing seismic reliability analysis. Lıeu et al. [15] found that deep neural network-based surrogate models can effectively evaluate limit state functions, as demonstrated through various examples involving multiple failure modes. Moreover, machine learning models could be integrated with the Bayesian probability framework, forming a probabilistic surrogate model robust against unwanted noises, as demonstrated in [16]. Zhou and Pen [17] combined the deep learning model with Gaussian Process and the probability density evolution method to tackle high-dimensional reliability problems. The proposed method is applicable for both static and dynamic systems. However, a drawback of the deep learning-based surrogate model is the necessity of preparing in advance a considerably large amount of data for adequately training the surrogate model, which undermines its practicality in real scenarios.

A promising advantage of the machine learning-based surrogate model is its ease of integration into existing frameworks without requiring a specific program like FEM or expert knowledge like statistical methods. Based on this property, this study proposes a active learning framework seamlessly integrating SS with a robust machine learning technique, namely ensemble learning, to improve further the efficiency of the standard SS method while alleviating the computationally expensive data preparation step. Table 1 compares recent works on active learning for structural reliability analysis, via which the distinctive contributions of the proposed approach are clarified. It can be seen that different methods combining Kriging with MC [18] have been developed to improve one or some aspects to achieve higher efficiency and accuracy. However, in the authors' opinion, these methods have two major limitations when working with extremely small failure probabilities. First, the Kriging based surrogate models are based on a subjective distance metric to assess the correlation between input points. This approach is not optimal for problems involving a high number of random variables, especially when they have different physical natures, which potentially impedes the Kriging model's performance. Second, for extremely small failure probabilities, the data is highly imbalanced; thus, the refinement process of Kriging-based methods may require a large number of iterations to get enough relevant data samples from failure regions for properly training the surrogate model. In order to remedy the first drawback, this study develops an ensemble learning model that stacks various classification machine/deep learning models to improve the final performance. Different models exploit different features from input data, perform chains of reasoning in different ways, and suffer different errors. Hence, the ensemble model is more robust and achieves higher or at least equal accuracy than any individual model. In order to address the second drawback, two techniques are leveraged: i) the SS strategy for progressively drawing relevant samples from intermediate sub-failure domains, ii) the sampling weighting technique, which assigns higher importance weights for scarce samples from failure regions than for normal samples, allows for mitigating the adverse effect of imbalanced data. In short, one summarizes the main contributions of this study as follows:

- A novel, accurate and efficient framework using adaptive ensemble learning paired with subset simulation, dubbed a-eSS, for analyzing the reliability of struc-

tures with a high number of random variables is designed and fully implemented without the need to prepare a FEM database in advance. The random variables considered in this study belong to structural parameter/variable uncertainties [16], which include material properties, cross-section dimensions of load bearing members, boundary condition stiffnesses, and external load intensities. Their values are drawn from predefined distributions with given statistical characteristics such as mean and standard deviation.

- The viability and performance of the a-eSS method are quantitatively illustrated through three case studies with increasing complexity. The computation results show that the proposed method outperforms competing approaches in terms of efficiency, i.e., 1.7 and 494 times faster than the SS and MC methods, respectively, while still providing highly similar reliability results.
- The informative insights about the mechanism of a-eSS are clarified through ablation, comparison, and parametric studies, thus increasing its applicability for further studies.

The remainder of the paper is organized as follows: Section 2 introduces the general workflow of the a-eSS framework, the theoretical background, the surrogate model using ensemble learning, and six machine/deep learning components. Section 3 presents three case studies involving a 15-bar truss structure, a 2D frame structure, and a 3D building structure. Finally, the conclusions, limitations, and some ideas for future works are drawn in Section 4.

## 2. Ensemble model for structural reliability analysis

Figure 1 illustrates the working flow of the proposed a-eSS approach (on the rightmost) and competing approaches, including the conventional Monte Carlo and Subset Simulation methods. The typical first step consists of defining key random variables that considerably impact structures' responses, for example, materials' mechanical properties, section sizes, load intensities, etc. These variables are described by predefined probability distributions and statistical characteristics such as mean and coefficient of variance values. For the conventional MC method, one first draws a large number of samples, then a series of FEM simulations with these samples will be carried out. The output of simulations will be tested with a problem-specific limit state function, such as a maximum allowable displacement, to calculate the failure probability and reliability results. Moreover, the reliability curve can be plotted, showing the evolution of the reliability index in function of various safety thresholds. The leftmost panel of Fig. 1 schematically illustrates these steps of MC. In order to reduce the number of simulations, the sampling strategy SS, which divides the failure region into a set of sub-failure domains, is utilized. The middle panel of the figure illustrates the realization steps of SS. Instead of drawing many samples, SS only considers a subset of samples within sub-failure domains and performs FEMs with these samples. The obtained outputs are then used to compute new intermediate safety thresholds and new sub-failure domains. After that, one employs the Modified Metropolis-Hastings (MMH) algorithm to draw new candidates. The latter needs to be checked to see whether they are within sub-failure domains.

Note that the checking process in the standard SS method is realized still using time-consuming FEM simulations. The simple yet effective idea of the proposed a-eSS model is to replace FEM with a surrogate model when checking the relevance of candidates generated by the MMH algorithm. Another noteworthy point of a-eSS is that it is fine-tuned by directly using samples of intermediate subsets of SS. In other words, it does not require preparing in advance a dataset for training the surrogate model, as usually done in some reviewed works. As illustrated in the rightmost panel, a-eSS only adds two more steps compared to the standard SS method. By doing so, a-eSS possesses double advantages: reducing the number of calculation samples (by SS) and shortening the checking time of new candidates (by the surrogate model), making it highly efficient. In the next paragraph, the details of two key components of a-eSS, i.e., the adaptive ensemble learning model and subset simulation, are described in detail.

The surrogate model, which can quickly identify the actual state of the structure based on structural parameters, excitation, and safety threshold, is one of the most critical components of the proposed framework. Its identification accuracy has a defining impact on the final performance. In order to achieve high accuracy, in this study, one leverages the ensemble learning method [28], which is a meta approach combining the outputs from multiple models, in other words, a model of models. Per the ensemble theory, an ensemble tends to provide better results with lower variance than any individual model used in the ensemble. This is because ensemble learning increases the diversity and decreases the risk of being stuck in local optima, potentially encountered by a single model.

Considering a dataset $D = \{X_i, y_i\}$ with $X_i$ being a vector of input features, $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]$, $m$ is the total number of features, $y_i$ is a binary value indicating the state of the structure, $y_i = 0$ means a safe state, $y_i = 1$ indicates a failure state. Let's denote $f(X_i)$ as an individual machine learning model. There are two major ways to combine $k$ different models $f_1(X_i), f_2(X_i), \dots, f_k(X_i)$: homogeneous and heterogeneous ensemble models. The former uses a set of models of the same type, while the latter employs models of different types. In this study, one works with imbalanced data as samples belonging to the failure domain occupy only a small probability; it is expected that increasing the diversity may be beneficial for such a scenario. This is why the heterogeneous, a.k.a, stacking ensemble model is selected, whose formula can be written as follows:

$$\hat{y}_i = G[f_1(X_i), f_2(X_i), \dots, f_k(X_i)], \tag{1}$$

where $G$ is an aggregation function.

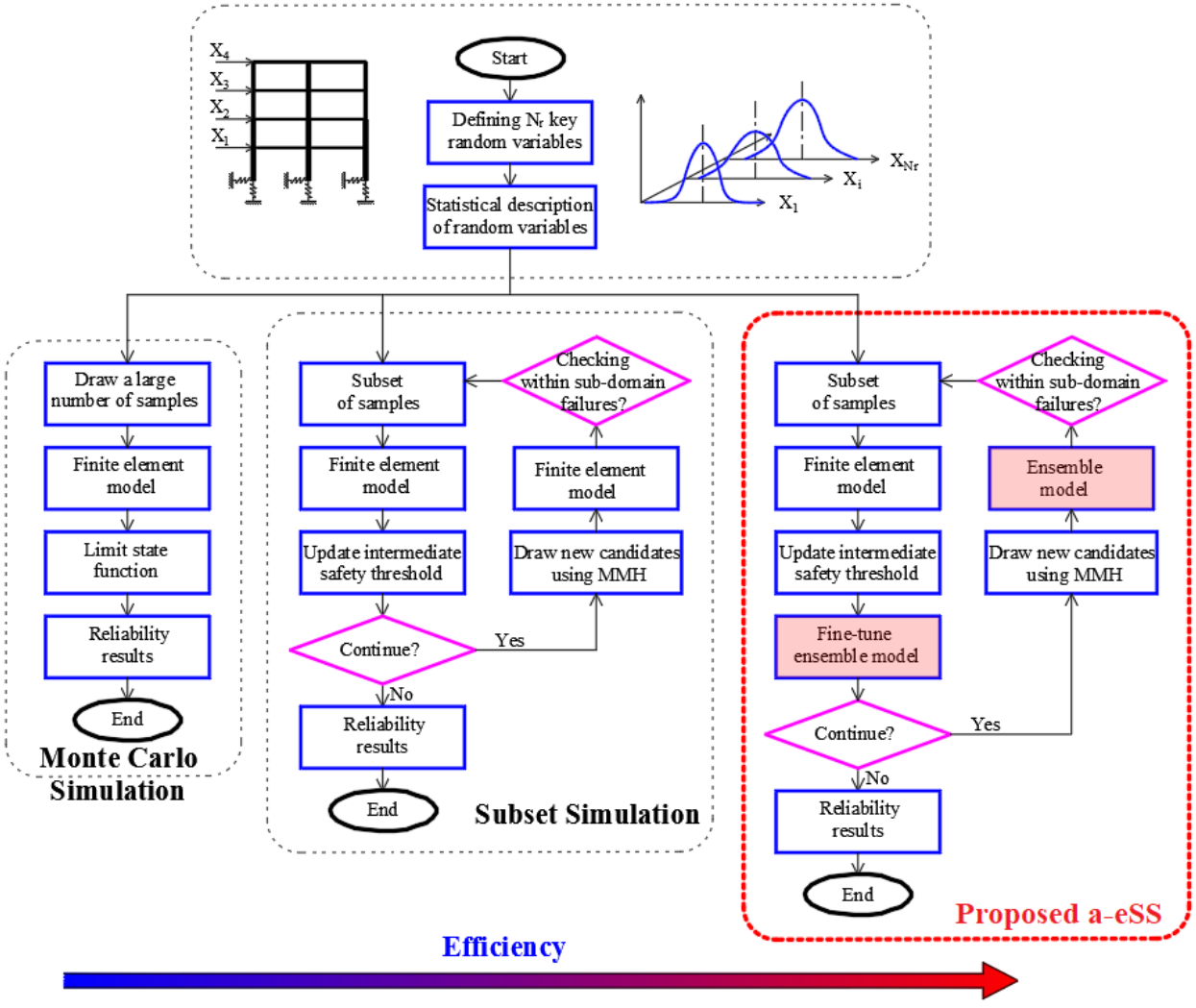



**Figure 1:** Working flow of the proposed a-eSS method (rightmost), Subset Simulation (middle), and conventional Monte Carlo Simulation (leftmost).

## 2.1. Model selection

Consider a training dataset $(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)$, where $X_i$ and $y_i$ are input feature vectors and the structure's condition (normal/failure) of the $i^{th}$ sample, respectively, it is required to identify the corresponding structure's condition for a new input $X^*$. Until now, a spectrum of ML classification algorithms has been developed. In order to construct an effective ensemble model, it should select different machine learning algorithms using different strategies for processing features rather than using algorithms in the same family to enrich the model diversity. On the other hand, it should prioritize popular, reliable, and easy-to-implement algorithms to promote their practical applicability. In contrast, using advanced machine learning models, for example, Random Forest (bagging), XGBoost (Boosting), etc., would add more tunable parameters, potentially making the ensemble model impractical. That is why one selects six different low/moderate complexity models, namely Logistic Classification, Naïve Bayes Classifier, Support Vector Machine, Decision Tree, K-Nearest Neighbors, and Multiple Layer Perceptron. They are among the most machine learning models widely acknowledged and used in different domains in both the academy [29] and industry [30**?** ].

The first machine learning model is the polynomial logistic classification model [31], thanks to its clarity and popularity. A polynomial combination of features is calculated before going through a nonlinear sigmoidal function for calculating probabilities of structures' possible states, as follows:

$$\hat{y^*} = w_0 + w_1 x_1^* + w_2 x_2^* + w_3 x_1^* x_2^* + w_4 (x_1^*)^2 + w_5 (x_2^*)^2 \quad (2)$$

where $w_i$ are coefficients of the polynomial function. The polynomial combination could include high-order powers of variables and their interactions if the boundary between classes is nonlinear. However, not all nonlinear relationships could be described by polynomial functions; in such a scenario, high-order terms may incur unnecessary complexity in the model, especially with high-dimensional inputs. Herein, the

quadratic polynomial function is selected.

The second model is the support vector machine model (SVM) [32], which is built upon a solid statistical learning framework. The core idea of SVM is to construct a linear hyperplane for separating data. Data samples belonging to the same side of the hyperplane are classified into the same classes. The optimal hyperplane would maximize the margin between samples of two classes on both sides of the hyperplane. In other words, this maximizes the distance between the hyperplane and the nearest data. Herein, the distance is obtained via an inner product. For nonlinear problems, if a linear hyperplane does not exist on the original feature spaces, a kernel function will be used to transform samples into higher-dimensional feature spaces where a linear decision boundary can be found. Mathematically, the SVM hyperplane can be described by:

$$h(X) = \sum_{i=1}^{N} w_i K(X, X_i) + b \tag{3}$$

where $K(X, X_i)$ is a kernel function, $b$ is an intercept term, $w_i$ are weight coefficients. One of the most used kernel functions is the Gaussian Radial Basis function, a.k.a, RBF, whose formula is written as below:

$$K(X_i, X_j) = \exp(-\gamma |X_i - X_j|^2), \tag{4}$$

where $\gamma$ is an user-defined parameter.

Another adopted model is the Naïve Bayes Classifier [33] which is an easy-to-implement classifier with a clear theoretical background and can provide quick results even when the input dimensionality is high. This model is based on the Bayes theorem and a relatively naïve assumption, as suggested by its name, that input features are independent of each other, i.e., there is no correlation between features. Let's denote $P(C_k|x_{i,1}, x_{i,2}, \dots, x_{i,m})$ is the probability of the class $C_k$ given a set of features $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]$. Per the Bayes' theorem, this probability can be calculated by:

$$P(y|x_{i,1}, x_{i,2}, \dots, x_{i,m}) = \frac{P(y)P(x_{i,1}, x_{i,2}, \dots, x_{i,m}|y)}{P(x_{i,1}, x_{i,2}, \dots, x_{i,m})} \tag{5}$$

By using the independence assumption, the conditional joint probability on the right-hand side of Eq. (5) is obtained as the product of a series of conditional probabilities, and the equation is rewritten as:

$$P(y|x_{i,1}, x_{i,2}, \dots, x_{i,m}) = \frac{P(y)\prod_{j=1}^{m} P(x_{i,j}|y)}{P(x_{i,1}, x_{i,2}, \dots, x_{i,m})} \tag{6}$$

Because probability $P(x_{i,1}, x_{i,2}, \dots, x_{i,m})$ is the same, thus the predicted class will be the one that maximizes the numerator of the right-hand side in Eq. (6):

$$y_i = \underset{k\in 1,\dots,K}{\operatorname{argmax}} P(C_k) \prod_{j=1}^{m} P(x_{i,j}|C_k) \tag{7}$$

Marginal probability $P(C_k)$ and each conditional probability $p(x_{i,j}|C_k)$ can be calculated from the training dataset.

Next, the fourth model is an interpretable learning method, namely, Decision Tree [34]. It can be expressed as an if-then clause-based model, using features for dividing data in a top-down fashion from the root node to branch nodes and leaf nodes. The root node corresponds to the whole data, while the leaf nodes correspond to prediction results. Let's introduce the entropy as a measure of the randomness calculated from class probabilities as follows:

$$\text{Entropy} = \sum_{k=1}^{K} -p_k \log_2 p_k, \tag{8}$$

where $p_k$ is the predicted probabilities for class $k$, entropy values range from 0 to 1. A low entropy value (near 0) means data samples within a node are highly similar, while a high entropy value (near 1) means samples are highly different and require further splitting. Next, intermediate branch nodes are divided such that the Information Gain value (IG) is maximized, where IG signifies the reduction in entropy as below:

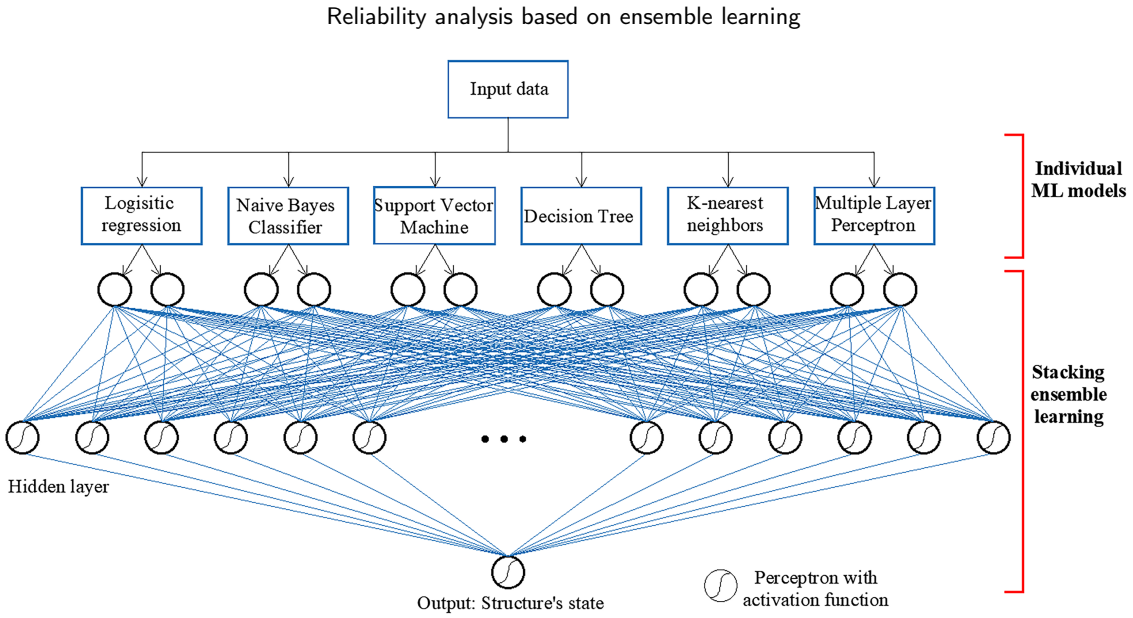
$$\text{IG} = \text{Entropy(parents)} - \text{Entropy(children)}, \tag{9}$$

in which Entropy(parents) and Entropy(children) are the entropies of parent and children nodes.

The fifth model is a non-parametric model, named K-Nearest Neighbor (K-NN) [35], i.e., no trainable weights need to be determined through a training process. The algorithm first calculates the distance between new sample $X^*$ with other data samples in the training dataset; then, its $K$ nearest neighbor samples are selected. After that, $y^*$ is derived based on the majority vote rule of these $K$ neighbors. The distance between data samples is usually the Manhattan or Euclidean distances. Standardization or normalization is carried out before the distance calculation if data features have different physical meanings and are of different scales. For K-NN, selecting an adequate value of $K$ is critical, which has an important impact on the performance of K-NN. A small value of $K$ will increase the variance of the model, while a high value of $K$ will reduce the model variance but give rise to bias. Usually, the cross-validation and elbow techniques are used to plot the performance of K-NN against different values of $K$. In this study, $K$ is set to 10 unless otherwise stated. It may not be an optimal value, but with the help of diversity, the ensemble model could still achieve high performance, as will be demonstrated later. To sum up, the steps of K-NN are as follows:

- Calculate distance between $(X^*, X_i)$ with $i = 1, \dots, N$.
- Sort these distances in ascending order and then select the first $K$ samples
- Identify the most frequent class from these chosen $K$ samples and assign it to $y^*$.

The last model is a multiple layer perceptron (MLP) [36] network that can extract non-linear relationships between features through its hierarchical architecture. The deeper the




(a) Working flow

```python
# Define individual models
def define_model(input_dim, output_dim):
    models = dict()
    models['lr']    = LogisticRegression()
    models['knn']   = KNeighborsClassifier()
    models['cart']  = DecisionTreeClassifier()
    models['svm']   = SVC(probability=True)
    models['bayes'] = GaussianNB()
    models['mlp']   = MLP_model(input_dim, output_dim)
    return models

# Define ensemble layer
def define_ensemble_model(input_dim, output_dim, hidden_dim):
    ensemble_model = Sequential()
    ensemble_model.add(Dense(hidden_dim, input_dim, activation='relu'))
    ensemble_model.add(Dense(output_dim, activation='softmax'))
    opt = tf.keras.optimizers.Adam(learning_rate=0.001)
    ensemble_model.compile(loss='categorical_crossentropy', optimizer=opt)
    return ensemble_model

# Stacking ensemble learning
def stacking_Emodel(models, ensemble_model, X,y):
    X_e = []
    for model in models.items():
        X_e.append(model.predict(X))

    X_e = np.concatenate(X_e, axis=1)
    ensemble_model.fit(X_e, y)
    return ensemble_model
```

(b) Code snippets written in Python

**Figure 2:** Representation of the stacking ensemble model.

architecture, the more complex relationship can be captured. However, a deeper architecture will increase the number of network weights, just requiring a larger volume of data. Otherwise, the overfitting problem may happen, where the model achieves high performance on training data but low accuracy on unseen data. Given a $d$-dimensional input, one selects a MLP network with a configuration of $[d, 3d, 3d, 2]$, i.e., there are two hidden layers, each having $3d$ neurons. Such a configuration is preliminarily evaluated by the authors; it can provide a good balance between the model complexity and performance in this study.

## 2.2. Stacking ensemble learning

Each machine learning model $f_j(X_i)$ can directly predict a class (hard voting) or provide a vector of probabilities for each class (soft voting). For the case of hard voting, $G$ is usually a maximum function, and the final predicted class is the one with the most votes from models. However, this way may not be optimal because a weak model has the same impact on the final results as a strong one. Hence, the soft voting scheme is selected where the probabilities for each class will go through a one-hidden-layer feed-forward network for predicting the final prediction. Considering model $f_j(X^i)$, the probabilities of $f_j(X_i) = 0$ and 1 are denoted by

$p_i$ and $q_i = 1 - p_i$, respectively. The aggregation function $G$ is rewritten as follows:

$$G[f_1(X_i), f_2(X_i), \dots, f_k(X_i)] = F_{out}\left[F_h(p_1, q_1, \dots, p_k, q_k, W_h), W_{out}\right] = \begin{bmatrix} p^i_{ens} \\ q^i_{ens} \end{bmatrix} \quad (10)$$

where $W_h$ and $W_{out}$ are trainable parameters of the hidden layer $F_h$ and output layer $F_{out}$, $p_{ens}$ and $q_{ens}$ are class probabilities predicted by the ensemble model. The number of neurons in the hidden layer is set to 64, which provides satisfactory results for the examples investigated in this study. The cross-entropy loss function is utilized to measure the classification performance of the ensemble model:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left(\log p^i_{ens} + \log q^i_{ens}\right) \quad (11)$$

The graphical presentation of the proposed stacking ensemble model is illustrated in Fig. 2 where the first part involves defining individual machine learning models, and the second part combines the output of ML models through a feed-forward network. Correspondingly, the code snippets of the ensemble learning written in Python are briefly presented in Fig. 2(b).

### 2.3. Subset Simulation

Considering a vector of independent random variables $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]$, where $x_{i,j}$ denotes a random variable representing uncertainty in structural parameters and load intensities with a predefined probability distribution $p_j$, with $j = 1, \dots, m$. Per [37], the joint probability of $X$ is computed as the product of $p_j(x_{i,j})$ as follows:

$$p(X_i) = \prod_{j=1}^{m} p_j(x_{i,j}). \quad (12)$$

A limit state of the structure is expressed via function $F(X_i)$. $F(X_i) \geq 0$ corresponds to a safety state; otherwise, if $F(X_i) < 0$, the failure happens. The probability of the failure $P_f$ is calculated as follows:

$$P_f = P[F(X) < 0] = \int \cdots \int_{F(X)<0} p(X)\, dx. \quad (13)$$

Next, one can calculate the reliability index ($RI$) as below:

$$RI = -\Phi^{-1}(P_f), \quad (14)$$

where $\Phi^{-1}$ is the inverse of the standard normal cumulative distribution function. The right-hand side of Eq. (13) can be numerically approximated by using MC simulation with a large number of samples drawn from predefined distributions. Let's introduce a binary function $I$ as follows:

$$I[f(X_i)] = \begin{cases} 0 & \text{if } F(X_i) \geq 0 \text{ (safe)}, \\ 1 & \text{if } F(X_i) < 0 \text{ (failure)}. \end{cases} \quad (15)$$

The probability of failure $P_f$ is then calculated as an expectation of the indicator function as below:

$$P_f = \frac{1}{N_{sample}}\sum^{N_{sample}} I[F(X)], \quad (16)$$

where $N_{sample}$ stands for the total number of samples. $N_{sample}$ needs to be sufficiently large to accurately estimate $P_f$. A reasonable value of $N_{sample}$ could be estimated by [38]:

$$N_{sample} = \frac{1 - \hat{P}_f}{CoV^2 \times \hat{P}_f}, \quad (17)$$

where $CoV$ denotes a coefficient of variance of $P_f$. Apparently, a very small $P_f$ would demand a large $N_{sample}$. If the estimation of $F(X_i)$ is time-consuming, the reliability analysis procedure with a large number of calculations will be computationally expensive or even prohibitive.

To reduce the number of calculation, the Subset Simulation method divides the failure domain $F$ into a sequence of sub-failure domains $F_1 \supset F_2 \supset \dots \supset F_{N_{sub}} = F$, with $N_{sub}$ being the total number of sub-domains. For each sub-domain, the failure probability is large enough to be acceptably approximated by using a feasible number of calculations with MC. Note that the sub-domains are hierarchically dependent, i.e., to compute $F_{s+1}$, one only considers samples within the sub-domain $F_s$. Thus, the obtained failure probability of sub-domain $i$ is a conditional probability, denoted by $P_f(F_{s+1}|F_s)$. By multiplying these conditional probabilities together, the final failure probability is obtained as follows:

$$P_f = P_f(F_{N_{sub}}) = P_f(F_1)\prod_{s=1}^{N_{sub}-1} P_f(F_{s+1}|F_s). \quad (18)$$

An important step of the SS method is how to sample data within sub-domain $F_s$ when calculating $P_f(F_{s+1}|F_s)$. Towards this end, a variant of the Markov chain Monte Carlo method, namely, the random walk modified metropolis hastings algorithm, is employed owing to its simplicity, flexibility, and effectiveness in sampling data from an arbitrary probability distribution.

In this algorithm, one predefines in advance a value of probability failure for each intermediate level, $P_f^{int}$. For level $s$ of SS, one computes the next threshold $F_{s+1}$ based on data samples and $P_f^{int}$. More specifically, if there are $N_s$ known samples $[X_1, \dots, X_{N_s}]$ within sub-domain $F_s$ sorted in ascending order: $F(X_{N_s}) > \cdots > F(X_1) > F_s$, then the threshold $F_{s+1}$ is defined as $F_{s+1} = f(X_{N_t})$ with $N_t = N_s \times P_f^{int}$.

For the next subset level $s + 1$, it is necessary to sample $N_s$ data within the critical region $F_s$ using $N_t$ known samples of the subset level $s$ determined above as seeds plus random walk quantities. In parallel, the threshold $F_s$ and these data samples $[X_1, \dots, X_{N_s}]$ are used to update the ensemble learning model such that it can accurately separate samples within or outside the sub-failure domain $F_s$. Besides, the sampling weighting technique is used to reduce

the unbalance problem of data, i.e., the number of samples corresponding to the normal state of the structure is considerably larger than that corresponding to the failure state. This technique assigns higher weights for the latter, making the training process achieves a faster convergence rate and higher final performance. Specifically, the ratio between the weight of samples belonging to the normal state and that of the failure state is 1:3.

Let's consider a feature $x_{i,j}^s$, of a sample data $i$ in the sub-domain $s$: $X_i^s = [x_{i,1}^s, x_{i,2}^s, \dots, x_{i,m}^s]$ with $i \in [1, N_t]$. For simplicity, in the rest of the section, one omits the superscript $s$. First, a proposal univariate Gaussian probability density function $q_j$ with a mean value equal to $x_{i,j}$ and standard deviation $\sigma_j$ is introduced. Next, a proposal value $\eta$ of component $j$ of the new sample is drawn according to a Gaussian probability density function as follows:

$$q_j^*(\eta|x_{i,j}) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(\frac{-(\eta - x_{i,j})^2}{2\sigma_j^2}\right), \tag{19}$$

The value of $\sigma_j$ is defined based on the original distribution $p_j(x_{i,j})$ and/or of $N_t$ seed samples. Afterward, a so-called acceptance ratio between the new sample $\eta$ and $x_{i,j}$ is calculated based on the given probability $p_j$ by:

$$r = \frac{p_j(\eta)}{p_j(x_{i,j})}. \tag{20}$$

then $\eta$ is accepted, i.e., $x_{i+1,j} = \eta$ with a probability of $min(1, r)$. Otherwise, $\eta$ is rejected and $x_{i+1,j} = x_{i,j}$.

Since each data sample is multi-dimensional, one applies the Metropolis-Hasting procedure separately for each feature, i.e., using $m$ proposal probability density functions for $m$ features. Such an approach is referred to as modified M-H or component-wise MH [39]. After that, the new sample $X^*$ is checked whether it lies within the sub-failure domain $F_j$. If not, one needs to regenerate $X^*$. Note that the checking is performed with the help of the updated, fast and accurate ensemble learning model. One repeats these above steps to generate $N_s$ samples for the sub-failure domain $F_s$. These samples are further sorted, and a new failure threshold $F_{s+1}$ is determined.

In short, this section has described how the ensemble learning and the Subset Simulation algorithm are combined in a complementary and synchronized way in the context of structural reliability analysis.

### 2.4. Training process setting

The training process settings of the ensemble model are specified as follows. The cross-entropy loss function is employed to measure the disparity between the ensemble model's outputs and the structure's actual conditions. To minimize the loss function, one employs the Stochastic Gradient Descent optimization algorithm, which gradually adjusts the model's weights in the opposite direction of the gradient of the loss function. The adjustment rate is characterized by the learning rate, which is initially set to 0.001, then reduced by a factor of 2 if the loss function on the validation data does not decrease for five consecutive epochs. The learning process finishes either when the learning rate decreases to $1 \times 10^{-5}$ or the maximum number of epochs is reached. The ensemble model is trained and validated by using the samples generated by the MMH algorithm in each sub-failure domain. These samples are randomly split into two datasets: training and validation datasets, with a widely used ratio of 8:2. Because there are multiple sub-failure domains; hence the ensemble model would be trained and retrained multiple times, accordingly. The model creation, training process, and evaluation of the a-eSS framework are implemented by the authors using the Python programming language and its scientific packages, including Numpy [40], Pandas for data preparation, Scikit learn [41] for machine learning models, Keras [42] for building MLP and stacking different models, and Matplotlib for data visualization.

## 3. Case studies

In this section, the applicability and performance of the proposed approach will be demonstrated via three case studies with increasing complexity. The same workflow as described in Fig. 1 is applied for all case studies, though the specific values of the ensemble model's hyperparameters and training settings will vary depending on the number of inputs and the problem complexity. In addition, parametric and ablation studies are also be carried out to gain insight into the mechanism of the a-eSS framework.

### 3.1. Case Study 1: 15-bar truss structure

The first case study is a simply-supported 2D Warren truss structure composed of 15 bars of the same cross-section. The truss has a total span of 1.2 m and a height of 0.26 m. Four concentrated loads are applied at the top joints, as shown in Fig. 3. The random variables include external load intensities, Young modulus, and section area, which are drawn from predefined distributions with mean and coefficient of variance values listed in Table 2. The output of interest is the vertical displacement of joint 3, located in the middle of the bottom chords of the truss.

The finite element model of the truss is created with the help of the open-source and reliable software OpenSees [43]. The truss bars are modeled by using linear elastic 2-node truss elements. Next, the proposed a-eSS method is adopted to calculate the reliability of the structure. The number of intermediate levels and the intermediate failure probability $P_f^{int}$ of SS are set to 9 and 0.3, respectively. Note that a too-small value of $P_f^{int}$ will require a large number of samples for each intermediate level, thus reducing the efficiency of SS. On the other hand, the larger the value of $P_f^{int}$, the more intermediate level is required to achieve a very small final failure probability, which potentially leads to higher accumulated errors. In this study, with $P_f^{int}$=0.3 and 9 intermediate levels, one can achieve a final failure probability of about $2 \times 10^{-5}$, i.e., a reliability index of around 4.1, and a relatively smooth reliability curve.

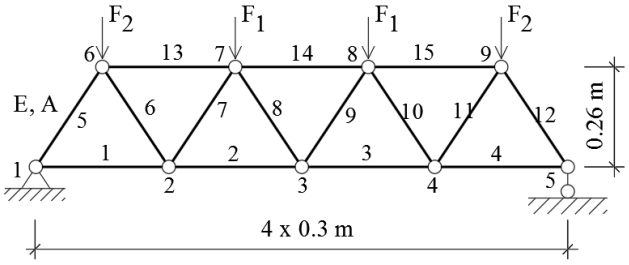



**Figure 3:** Schematic representation of the 15-bar truss structure.

**Table 2**
Random variables of the 15-bar truss structure

| Variable | $F_1$ (kN) | $F_2$ (kN) | $E$ (GPa) | $A$ (mm$^2$) |
|---|---|---|---|---|
| Mean | 10 | 20 | 210 | 113 |
| CoV | 0.1 | 0.1 | 0.05 | 0.1 |
| Dist. | Normal | Normal | Normal | Normal |

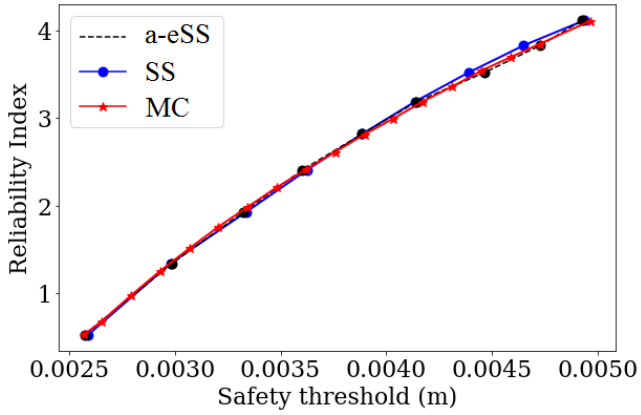



**Figure 4:** Reliability index curves for the 15-bar truss structures obtained by the a-eSS, SS, and MC methods.

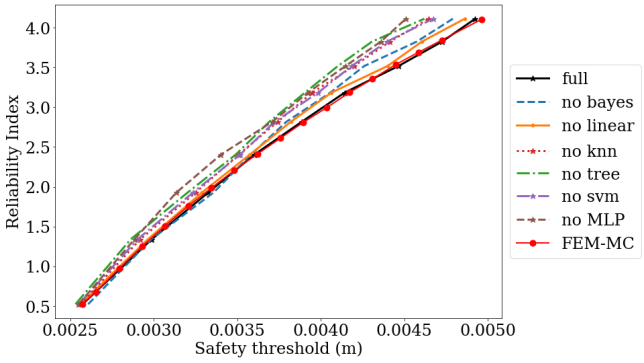



**Figure 5:** Effect of removing machine learning/deep learning components on the performance of the ensemble model.

In terms of the ensemble learning model, it is noted that the backbone of the proposed method is to leverage the diversity of different machine learning/deep learning models for increasing the classification performance rather than tuning a set of optimal hyperparameters. Therefore, for each ML/DL model component, its hyperparameters are set by default values. For SVM, the regularization parameters $C$ and $\gamma$ are set to 1.0 and $1/m$, respectively, with $m$ being the number of random variables. For the decision tree model, the minimum number of samples needed for a split is 10, i.e., a split is carried out when and only when there are more than ten samples at a node. For the MLP model, the model architecture is [5, 15, 15, 2], consisting of an input layer with five perceptrons (4 random variables plus a safety threshold) and two hidden layers; each layer has 15 perceptrons and an output layer with two perceptrons. For basic models such as linear regression and Naïve Bayes models, no special hyperparameters are required.

Once the a-eSS model is defined and trained, it is used to calculate failure probabilities for different safety thresholds, then derive the corresponding reliability indexes and plot the reliability curve. The computed reliability results for the 15-bar truss structure are depicted in Fig. 4. Moreover, the figure also presents the results obtained via the conventional Monte Carlo simulation using FEM and the subset simulation using FEM for comparison purposes. In the figure, the solid line with dot symbols, red line with star symbols, and dashed line with triangle symbols denote results provided by a-eSS, MC, and SS, respectively. It can be seen that a high agreement between these three methods is obtained, which reaffirms the correct implementation and credibility of the proposed a-eSS method. For MC, the required number of simulations is up to $3 \times 10^6$ to reach a reliability index of around 4.0, according to Eq. (17). Meanwhile, the number of calculations for SS is only $300 \times 9 = 2700$. Furthermore, the CPU time required by a-eSS for the whole reliability analysis of the truss structure is 23.7 minutes, while it takes 28.6 and 217.5 minutes with SS and MC, respectively. Therefore, a-eSS reduces the computation time by 1.2 and 9.2 folds compared to SS and MC for this case study. Note that a-eSS does not require performing an extensive series of FEM simulations in advance to prepare training data. It directly uses intermediate results to improve the ensemble model's performance.

In order to get more insights into the functioning of the ensemble model, a leave-one-out study is carried out to investigate the contribution of each machine learning component to the performance of the ensemble model. Specifically, one repeats the above reliability analysis multiple times, each time a machine learning component model is removed, while the others remain unchanged. The results of the leave-one-out study are illustrated in Fig. 5. It is noticed that removing the linear regression reduces the accuracy of the ensemble model slightly, as the corresponding curve (solid orange line with dot symbols) diverts from that of a-eSS by a small amount. In contrast, the effects of removing MLP and Decision Tree are profound, as deviations are significant, especially with high-reliability indices. For example, with a safety threshold of 0.45, the model without MLP overestimates the RI, i.e., RI=4.0 vs. 3.5 by a-eSS. Besides, the impact of SVM and Bayes are more important than the linear

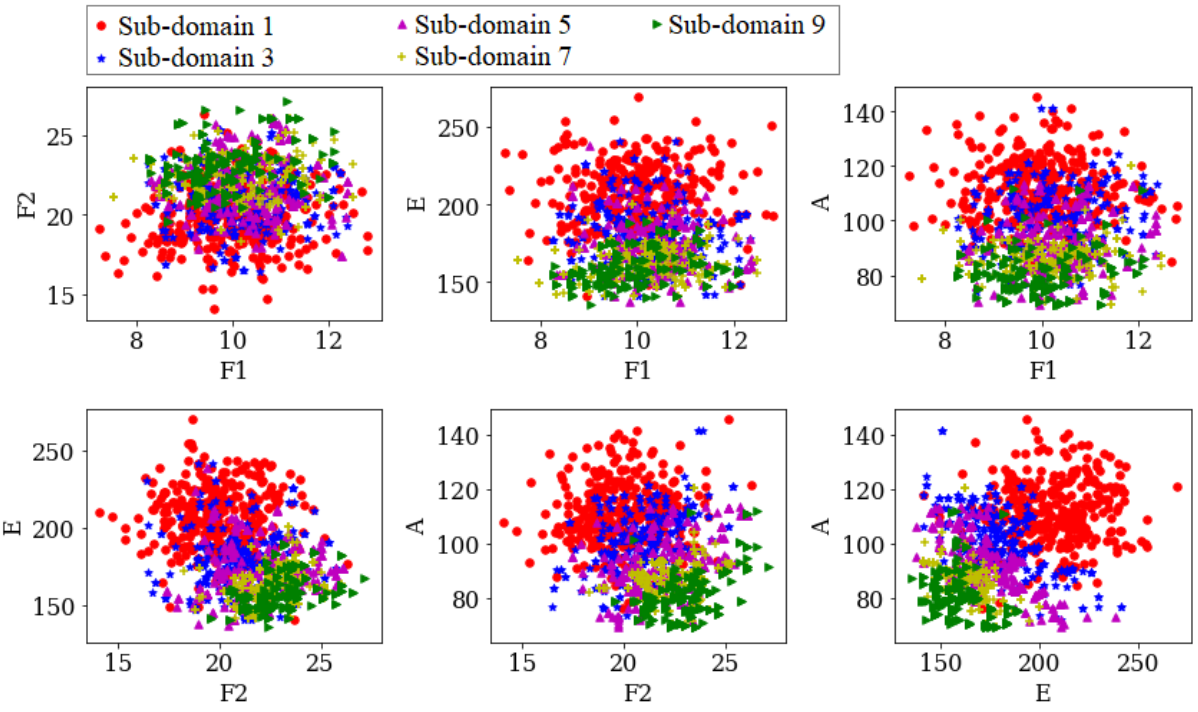



**Figure 6:** Random variables including $F_1, F_2, E$, and $A$ drawn at different intermediate levels of SS. The random variables are plotted against each other for better observation.

**Table 3**
Random variables of the 4-story 2-bay frame structure.

| Variable | $F_1$ (kN) | $F_2$ (kN) | $F_3$ (kN) | $F_4$ (kN) | $C$ (N/m$^2$) | $D$ (m) | $A_1$ (m) | $H_1$ (m) | $A_2$ (m) | $H_2$ (m) |
|---|---|---|---|---|---|---|---|---|---|---|
| Mean | 50 | 100 | 100 | 50 | 100000 | 2 | 1.5 | 0.6 | 1.8 | 0.6 |
| CoV | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Dist. | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal |

regression model, but less than those of MLP and Decision Tree.

Next, to explore how the SS strategy draws random variables $[F_1, F_2, E, A]$, these variables are plotted against each other for intermediate levels 1, 3, 5, 7, and 9, as shown in the subplots of Fig. 6. Apparently, at the first intermediate level, the random variables spread over a large domain; after that, they move towards some specific failure domains; for example, in Fig. 6, values of $E$ and $F_2$ progressively approach 150 and 23, at the ninth intermediate level.

In short, via a relatively simple yet representative structure with four random variables, the realization steps, potency, and mechanism of the proposed approach have been demonstrated. In the following subsection, a-eSS will be applied to more complex structures with more random variables.

### 3.2. Case Study 2: 2-bay 4-story frame structure

The second case study is a 2D reinforced concrete 4-story 2-bay frame structure with shallow foundations. The frame has a total height of 17 m, and its bay width is 6 m. The structure is subjected to four concentrated horizontal loads acting at the story levels. Each foundation is modeled by two

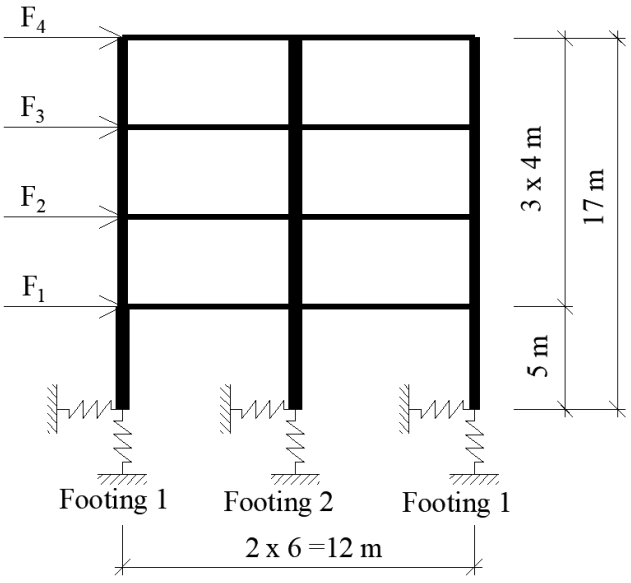
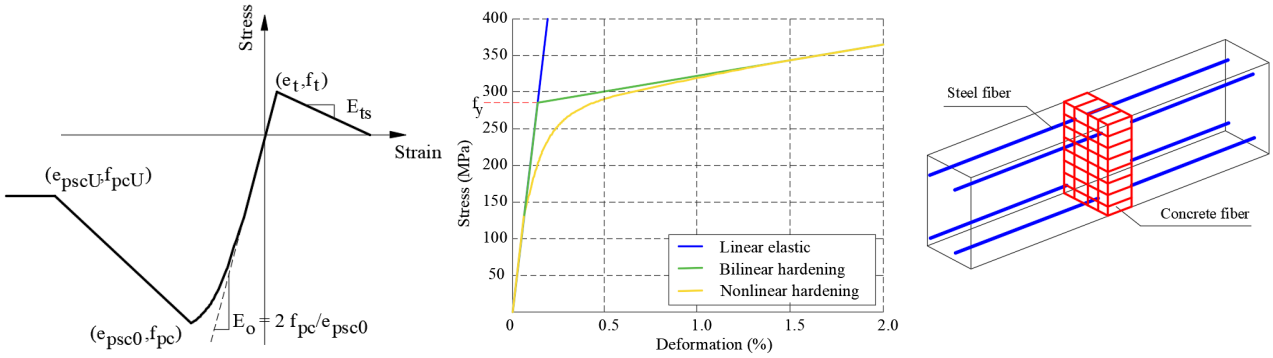



**Figure 7:** Schematic representation of the 4-story 2-bay frame structure.




(a) (b) (c)

**Figure 8:** Constitutive models for concrete (a), steel (b), and the fiber approach for modeling the RC section in the OpenSees software (c).

springs, i.e., vertical and horizontal ones, as schematically displayed in Fig. 7. The stiffnesses of the springs are derived from soil properties and foundations' geometry. More details about shallow foundation modeling can be found in the OpenSees documentation [44]. There are, in total, 10 random variables considered in this example which are the magnitudes of horizontal loads ($F_1$ to $F_4$), soil cohesion ($C$), depth of embedment of the footings ($D$), footing heights ($H_1$, $H_2$) and widths ($A_1$, $A_2$). The longitudinal reinforcement ratio for the columns and beams are 2% and 1%, respectively. The statistical values of these random variables are enumerated in Table 3. To model the elastoplastic behavior of concrete, the Kent-Scott-Park model [45] is adopted (Fig. 8(a)). For the compressive behavior, initially, the stress-deformation curve features a linear elastic relationship with a slope $E_0$=35 GPa; after attaining the maximum strength $f_{pc}$ = 30 MPa, the concrete strength diminishes while the deformation still increases until the crushing strain $e_u$. On the other hand, the tensile behavior is characterized by the tensile strength $f_t$. For the steel reinforcement, its behavior is described via a bilinear curve comprised of a linear elastic branch with slope $E_s$ and a hardening state with a smaller slope $E_1 = b \times E_s$, as illustrated in Fig. 8(b).

The fiber approach is utilized to simulate the reinforced concrete (RC) section. This approach can account for moment-curvature and axial force-deformation relationships simultaneously; thus, it can adequately model composite actions between concrete and reinforcement. The fiber approach divides RC sections into a set of fibers both horizontally and vertically, as illustrated in Fig. 8(c). Then, the response of the cross-section is obtained by summing the contributions of individual fibers. The output under investigation in this example is the top floor's horizontal displacement.

Next, an a-eSS model is engineered with nine intermediate levels, intermediate failure probability $P_f^{int} = 0.3$, number of samples in each sub-failure domain $N_s = 300$. For an intermediate level $s$, 300 numerical simulations with data samples within sub-domain $F_{s-1}$ are carried out using FEM at first, the obtained results are then used to determine safety

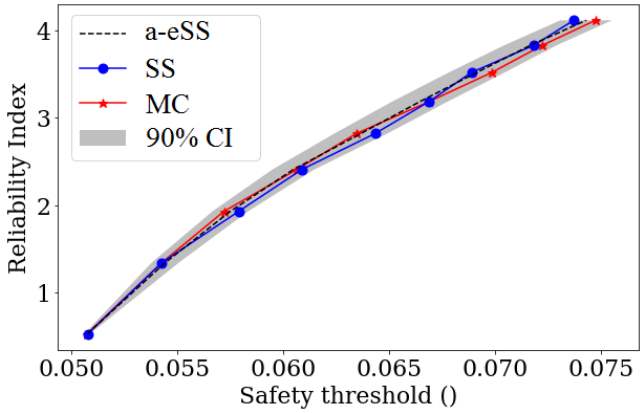
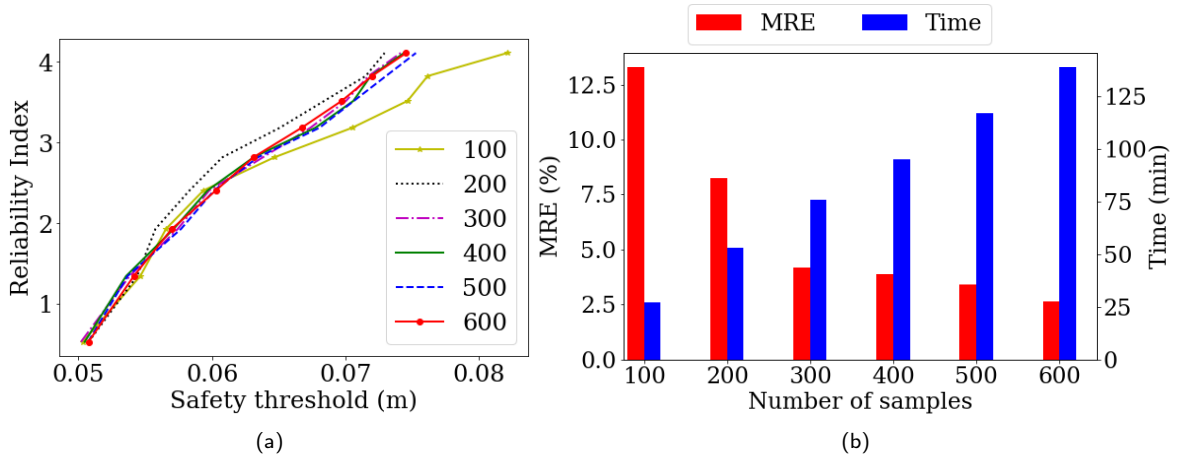



**Figure 9:** Reliability index curves for the 4-story 2-bay structure obtained by the a-eSS, SS, and MC methods.

threshold $F_s$ and then fine-tune the ensemble model. The updated ensemble model is utilized to check whether candidates generated by the MMH algorithm lie within the sub-failure domain $F_s$. The computed reliability results for the 2D frame structure are presented in Fig. 9 for three methods involving a-eSS, MC, and SS. Besides, one repeats 100 times the reliability analysis with the proposed framework a-eSS, then reports the confidence of obtained results via a 90%-confidence interval as shown by the grey area in the figure, providing additional information about the credibility of obtained results. Apparently, a-eSS yields results highly correlated with MC and SS as their curves are well aligned. Despite some slight discrepancies between the MC's curve and the mean curve of a-eSS, the former is still enclosed inside the 90%-CI area of a-eSS.

In order to assess the influence of the number of samples in each intermediate level ($N_s$) on the a-eSS performance, the reliability analysis is performed with different numbers of samples from 100 to 600, and corresponding results are highlighted in Fig. 10. The picture features RI curves (Fig. 10(a)), the CPU time and mean relative errors (MRE) (Fig. 10(b)). A low value of $N_s$ cannot provide sat-




(a) (b)

**Figure 10:** Effect of the number of data samples used in each SS intermediate level on reliability results and computational times.

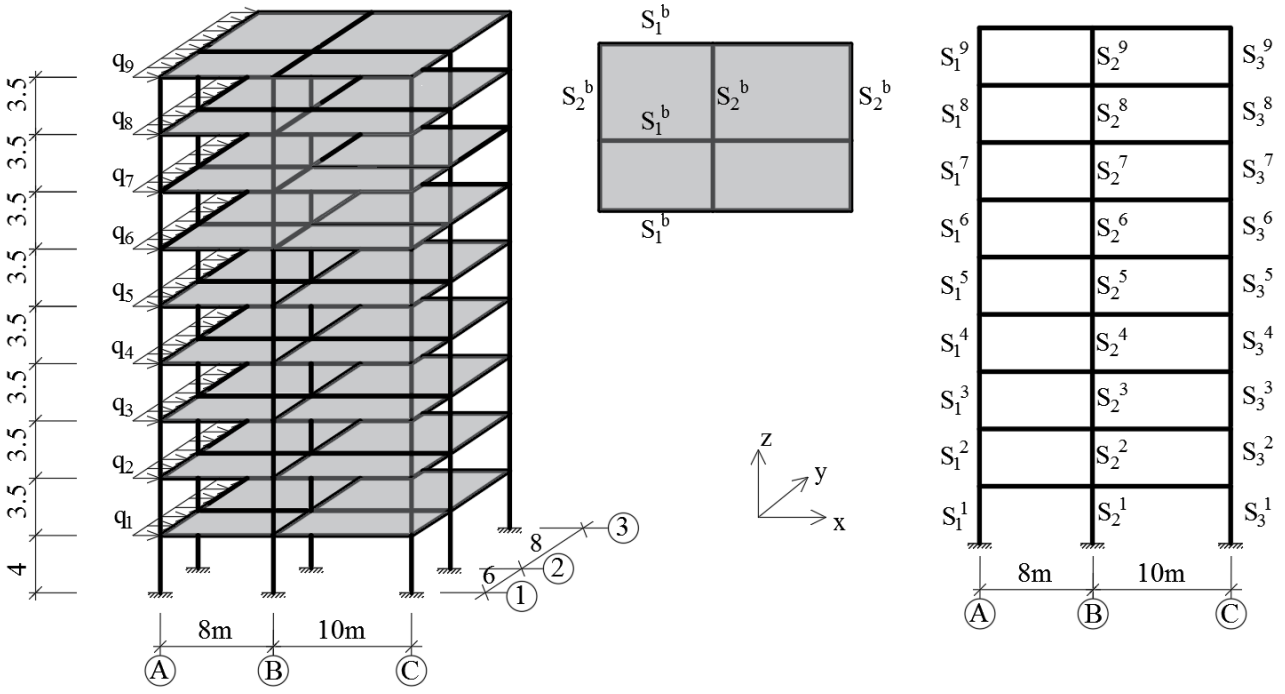



**Figure 11:** Schematic representation of the 9-story building structure.

isfactory results as the RI curves of $N_s$ = 100 and 200 are clearly different from those of higher $N_s$. More specifically, MRE decreased from about 13% for $N_s$ = 100 to around 1.2% for $N_s$ = 600. However, the CPU times increase proportionally with increasing $N_s$, e.g., CPU time of $N_s$=200 is nearly double that of $N_s$=100 (52 s vs. 27 s). Besides, $N_s$ = 300 provides a good balance between performance and computation complexity as its MRE is less than 5% while its CPU time is about half of that of $N_s$=600 (74 s vs. 135 s).

### 3.3. Case Study 3: 9-story building structure

The third case study considers a 3D nine-story two-bay building structure with up to 42 random variables, subjected to horizontal uniform line loads at each story as depicted in Fig. 11. The figure shows its 3D representation, floor plan, and front view with dimensions and assigned cross-sections. The typical story height is 3.5 m, except for the first story with a height of 4 m; thus, the structure's total height is 32 m. The bay widths of the structure in the X-direction are 8 m and 10 m, while the bay widths in the Y-direction are 6 m and 8 m. The longitudinal rebar ratios are 2% for the column and 1% for the tension area of the beam. The random variables

**Table 4**
Random variables of the 9-story building structure.

| Variable | $S_1^1$ (mm) | $S_2^1$ (mm) | $S_3^1$ (mm) | $S_1^2$ (mm) | $S_2^2$ (mm) | $S_3^2$ (mm) | $S_1^3$ (mm) | $S_2^3$ (mm) | $S_3^3$ (mm) |
|---|---|---|---|---|---|---|---|---|---|
| Mean | 400 | 400 | 400 | 400 | 400 | 400 | 400 | 400 | 400 |
| CoV | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Distribution | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal |
| Variable | $S_1^4$ (mm) | $S_2^4$ (mm) | $S_3^4$ (mm) | $S_1^5$ (mm) | $S_2^5$ (mm) | $S_3^5$ (mm) | $S_1^6$ (mm) | $S_2^6$ (mm) | $S_3^6$ (mm) |
| mean | 350 | 350 | 350 | 350 | 350 | 350 | 350 | 350 | 350 |
| CoV | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Distribution | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal |
| Variable | $S_1^7$ (mm) | $S_2^7$ (mm) | $S_3^7$ (mm) | $S_1^8$ (mm) | $S_2^8$ (mm) | $S_3^8$ (mm) | $S_1^9$ (mm) | $S_2^9$ (mm) | $S_3^9$ (mm) |
| mean | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |
| CoV | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Distribution | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal |
| Variable | $H_1^b$ (mm) | $H_2^b$ (mm) | $q_1$ (N/m) | $q_2$ (N/m) | $q_3$ (N/m) | $q_4$ (N/m) | $q_5$ (N/m) | $q_6$ (N/m) | $q_7$ (N/m) |
| mean | 600 | 500 | 4.7 | 5.1 | 5.4 | 5.6 | 5.8 | 5.9 | 6 |
| CoV | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Distribution | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal | Normal |
| Variable | $q_8$ (N/m) | $q_9$ (N/m) | $f_{DL}$ (N/m$^2$) | $f_{LL}$ (N/m$^2$) | $E$ (GPa) | $\rho$ (kg/m$^3$) | | | |
| mean | 6.1 | 6.2 | 3 | 2 | 30.0 | 2400 | | | |
| CoV | 0.2 | 0.2 | 0.1 | 0.1 | 0.05 | 0.1 | | | |
| Distribution | Normal | Normal | Normal | Normal | Normal | Normal | | | |

involve the intensities of horizontal loads, vertical dead load and live load, column cross-sections, beam heights, Young modulus, and concrete density. The statistical characteristics of these random variables, including the mean and CoV values, are listed in Table 4. Note that, with no loss of generality, the stories are supposed to have the same beam sections, and members of frames of axes 1, 2, and 3 are identical. The FE model of the structure is realized in OpenSees using concrete and steel constitutive laws described in example 2. The number of structural members in this example increases up to 189. Each column-beam joint has 6 degrees of freedom. The equation of equilibrium of the structure is numerically solved with the help of the Newton-Raphson algorithm. The solution convergence is tested by checking whether the norm of the displacement increments is smaller than a small enough tolerance. The limit state function of this example is constructed by comparing the maximum top floor horizontal displacement in the X-direction with a safety threshold.

Next, the input layer of a-eSS is modified to a layer of 43 perceptrons to account for 42 random variables plus a safety threshold value. Next, the a-eSS model is applied to analyze the structure's reliability using the same procedure as the previous examples. Obtained results are graphically presented via RI curves in Fig. 12. It is noticed that a-eSS can acceptably approximate the MC method, though for high

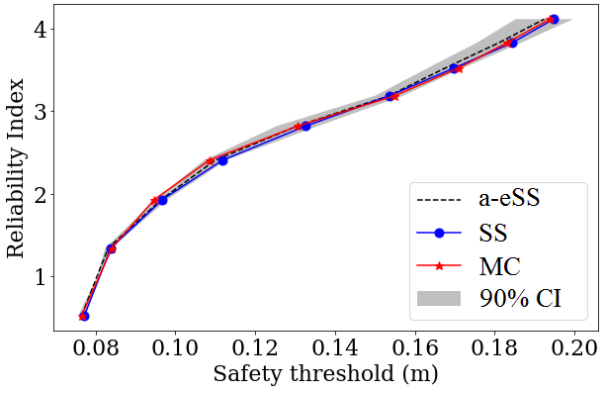



**Figure 12:** Reliability index curves for the 9-story building structure obtained by the a-eSS, SS, and MC methods.

RI (>3.2), the uncertainty of results increases, as illustrated by the widening of the 90%-CI area.

It is essential to ensure the correctness of a-eSS after each fine-tuning step; otherwise, intermediate errors would accumulate, leading to uncontrolled results. Fig. 13 demonstrates the accuracy of a-eSS for nine SS intermediate levels via confusion matrices. The 0/1 values on the X-axis of the confusion matrix denote the structure's actual condition (safe/failure), while the values on the Y-axis correspond to

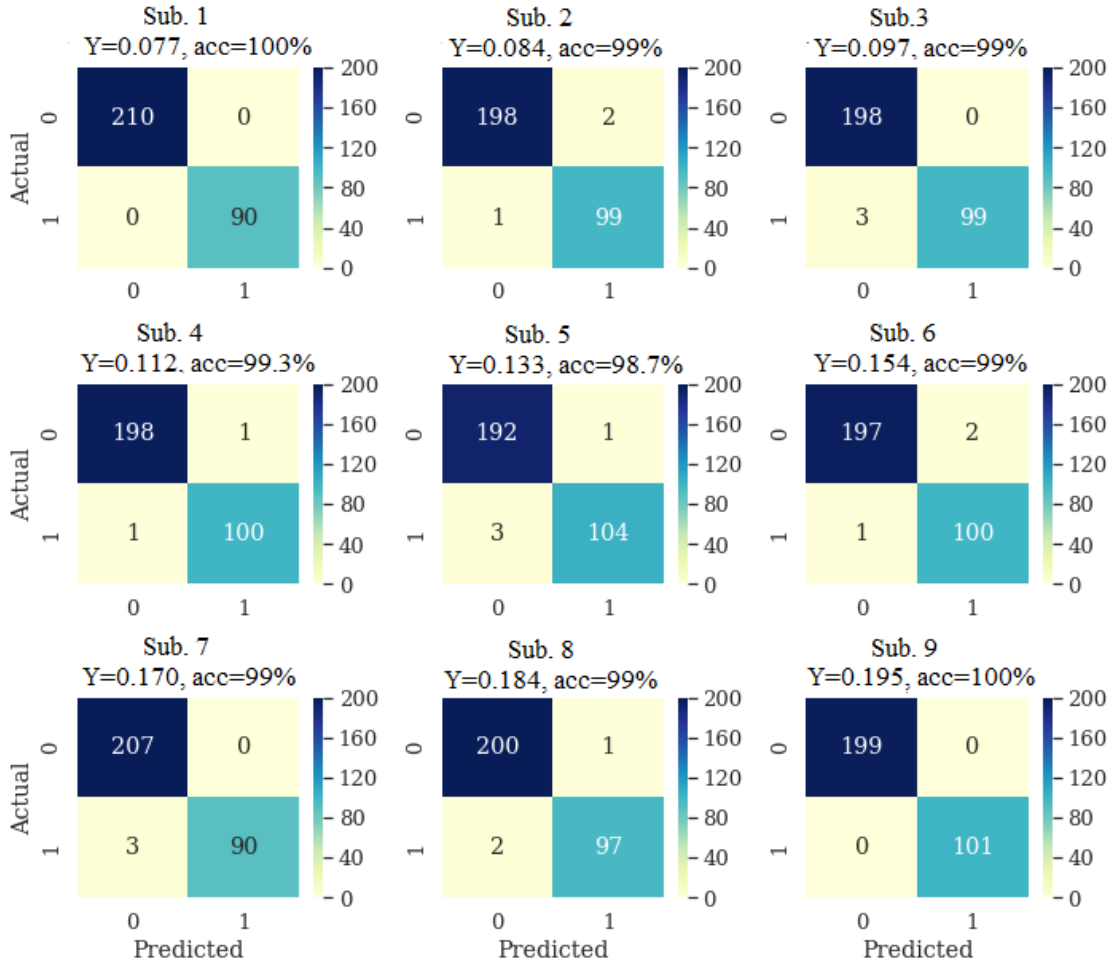



**Figure 13:** Confusion matrices representing the accuracy of the ensemble models for different SS intermediate levels.

**Table 5**
Comparison results of time complexity for reliability analysis of the 9-story building structure between a-eSS and counterparts.

| Model | Data preparation (min) | Training process (min) | Reliability analysis(min) | Total CPU times (min) |
|---|---|---|---|---|
| a-eSS | 0 | 8 | 64 | 72 |
| eSS | 360$^{(*)}$ | 5 | 56 | 421 |
| MC | 0 | 0 | 35565$^{(**)}$ | 35565 |
| SS | 0 | 0 | 120 | 120 |

$^{(*)}$ for $3 \times 10^4$ samples
$^{(**)}$ for $3 \times 10^6$ samples

predictions by a-eSS. The ratio between the sum of diagonal cells and the total number of samples will provide the proposed approach's accuracy. It is noticed that the ensemble model achieves high classification accuracies from 98.7% to 100% across all intermediate levels. Therefore, it is justifiable to use the ensemble model as an alternative and complementary to FEM when assessing the structure's condition.

In terms of computation complexity, Table 5 enumerates the CPU times of main computation steps for four methods, i.e., a-eSS, MC, SS, and a method, called eSS, using a pretrained ensemble model to replace FEM completely. The computation steps presented in the table include data preparation for training the ensemble model, the training process, and reliability analysis. Based on Eq. (17), for a RI value of approximately 4.0 and a CoV value of 0.1, the required number of samples for MC is about $3 \times 10^6$. Such a number is equal to or higher than those considered in some published works; for example, Zhao et al. [25] and Zhou and Peng [17] both performed MCS using $10^6$ samples for structures with 67 and 110 random variables, respectively. The estimated CPU time of the reliability analysis of the 3D building structure using MCS with $3 \times 10^6$ samples is up to 35565 minutes

on a single CPU (Table 5). Actually, calculations were operated in parallel on a computation server equipped with 12 Intel Xeon E5-2696 processors, 32 GB Ram, and a RTX3090 GPU, and the actual calculation time was reduced to around 59.3 hours. It can be seen that eSS required significant CPU times to prepare a training database, making its overall computation cost relatively high. Meanwhile, a-eSS is adaptively fine-tuned through each intermediate level of the subset simulation and does not require preparing data beforehand. On the other hand, compared to SS, the a-eSS method replaces the time-consuming FEM with the fast ensemble model when checking the relevance of candidates proposed by the MMH algorithm. That is why a-eSS is highly efficient, requiring only about 72 minutes to construct the whole RI curves, i.e., 1.7 times, 5.8 times, and 494 times faster than SS, eSS, and MC, respectively. In summary, this example qualitatively and quantitatively reaffirms the applicability of a-eSS to a complex 3D structure with numerous random variables.

## 4. Conclusion

This study develops an accurate and efficient active learning framework for structural reliability analysis, which is applicable to high-dimensional problems with extremely low failure probabilities. The backbone idea of the proposed framework is the seamless integration of a decent surrogate model based on ensemble learning with the SS sampling strategy. The paper describes different perspectives of the proposed a-eSS framework in detail, including the theoretical background, implementation steps, values of key hyperparameters, intermediate results, and ablation studies. With obtained computational and comparison results, some concluding remarks are drawn as follows

- The proposed ensemble learning-based surrogate model is capable of effectively and efficiently predicting the structure's responses. It achieves better or at least equal prediction accuracy to any individual machine learning model while not increasing the model complexity considerably. Hence, the surrogate model can be built, trained, and deployed with standard personal computers, facilitating its practical application.
- Unlike the deep learning counterparts, which require a large number of training data generated in advance via time-consuming FEM, the proposed a-eSS is adaptively trained with a sequence of small-size relevant datasets. Thus, the tedious data generation process is significantly alleviated.
- Integrating the surrogate model into SS is straightforward thanks to the improved portability of the modern machine learning model and by omitting a number of barriers such as license checking, pre/post-processing steps with graphical user interface, and communication between different softwares. This advantage allows for forming a smooth working flow for structural reliability analysis.
- The a-eSS framework brings two benefits to the computation of the failure probability: i) less number of calculations are needed, and ii) the computational cost for generating new relevant samples within the subfailure regions is significantly reduced. That is why a-eSS can be utilized to compute extremely small failure probability. Furthermore, a-eSS helps establish RI and failure probability curves, showing a full picture of the structure's reliability with different safety thresholds.
- All the aforementioned statements are qualitatively and quantitatively demonstrated through three case studies with increasing complexity. Especially for a complex 3D building structure with 42 random variables, the a-eSS approach reduces the computational time nearly 500 times with a mean relative error of less than 5% compared to MC.

The current version of the a-eSS framework can only address single failure criteria; thus, in the next step of the study, it is desired to extend a-eSS to handle multiple failure scenarios. This could be done by engineering a multitasking model which can provide different structural responses at the same time. Such a multitasking model will require a carefully designed compound loss function and corresponding joint training strategies. It is also required to extend the MMH algorithm to draw relevant samples within failure domains defined for multi-criteria. Another promising research direction is to account for time-varying inputs whose statistical characteristics could change over time. For this purpose, on the one hand, one resorts to sophisticated deep learning architectures specialized in handling time-varying data such as Long Short Term Memory, 1D Convolution Neural Network, or Transformer. On the other hand, some non-probability techniques, such as the ellipsoid mode and subinterval method [46] are adopted if unknown-but-bounded uncertain variables are considered. Such a framework is promising for addressing time-dependent reliability analysis-based applications such as dynamic topology optimization [47], intelligent design technologies [48], remaining useful life prediction, etc. Finally, it is noteworthy to test the viability of the proposed approach with a realistic structure. For this purpose, physical tests are first carried out to evaluate the current values of structural parameters. Next, these measured values are used to construct a calibrated numerical model. After that, the procedure described in this study is used to evaluate the structure's reliability thoroughly.

## Acknowledgement

This work was financially supported by the Hanoi University of Civil Engineering (Vietnam). The work of Huan Nguyen is supported by the VinGroup Innovation Foundation (VinIF) grant ID 'VINIF.2021.DA00192'.

## References


[1] I. Papaioannou, D. Straub, Combination line sampling for structural reliability analysis, Structural Safety 88 (2021) 102025.

[2] S. Xiao, S. Oladyshkin, W. Nowak, Reliability analysis with stratified importance sampling based on adaptive kriging, Reliability Engineering & System Safety 197 (2020) 106852.

[3] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic engineering mechanics 16 (4) (2001) 263–277.

[4] W.-C. Hsu, J. Ching, Evaluating small failure probabilities of multiple limit states by parallel subset simulation, Probabilistic Engineering Mechanics 25 (3) (2010) 291–304.

[5] D. A. Alvarez, F. Uribe, J. E. Hurtado, Estimation of the lower and upper bounds on the probability of failure using subset simulation and random set theory, Mechanical Systems and Signal Processing 100 (2018) 782–801.

[6] A. U. Ebenuwa, K. F. Tee, Fuzzy-based optimised subset simulation for reliability analysis of engineering structures, Structure and Infrastructure Engineering 15 (3) (2019) 413–425.

[7] A. Abdollahi, M. A. Moghaddam, S. A. H. Monfared, M. Rashki, Y. Li, A refined subset simulation for the reliability analysis using the subset control variate, Structural Safety 87 (2020) 102002.

[8] L. Katafygiotis, S. H. Cheung, K.-V. Yuen, Spherical subset simulation ($s^3$) for solving non-linear dynamical reliability problems, International journal of reliability and safety 4 (2-3) (2010) 122–138.

[9] I. Kaymaz, C. A. McMahon, A response surface method based on weighted regression for structural reliability analysis, Probabilistic Engineering Mechanics 20 (1) (2005) 11–17.

[10] G. Su, B. Yu, Y. Xiao, L. Yan, Gaussian process machine-learning method for structural reliability analysis, Advances in Structural Engineering 17 (9) (2014) 1257–1270.

[11] C. Zhou, N.-C. Xiao, M. J. Zuo, W. Gao, An active kriging-based learning method for hybrid reliability analysis, IEEE Transactions on Reliability.

[12] T. Simpson, F. Mistree, J. Korte, T. Mauery, Comparison of response surface and kriging models for multidisciplinary design optimization, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1998, p. 4755.

[13] H. V. Dang, R. Trestian, T. Bui-Tien, H. X. Nguyen, Probabilistic method for time-varying reliability analysis of structure via variational bayesian neural network, in: Structures, Vol. 34, Elsevier, 2021, pp. 3703–3715.

[14] M.-H. Ha, T.-P. Nguyen, D.-M. Hoang, V.-H. Dang, Reliability analysis of structures subjected to seismic excitation using a deep learning-based surrogate model, in: CIGOS 2021, Emerging Technologies and Applications for Green Infrastructure, Springer, 2022, pp. 1917–1926.

[15] 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.

[16] Y. Liu, L. Wang, K. Gu, M. Li, Artificial neural network (ann)-bayesian probability framework (bpf) based method of dynamic force reconstruction under multi-source uncertainties, Knowledge-Based Systems 237 (2022) 107796.

[17] T. Zhou, Y. Peng, Efficient reliability analysis based on deep learning-enhanced surrogate modelling and probability density evolution method, Mechanical Systems and Signal Processing 162 (2022) 108064.

[18] B. Echard, N. Gayton, M. Lemaire, Ak-mcs: an active learning reliability method combining kriging and monte carlo simulation, Structural Safety 33 (2) (2011) 145–154.

[19] A. A. Chojaczyk, A. P. Teixeira, L. C. Neves, J. B. Cardoso, C. G. Soares, Review and application of artificial neural networks models in reliability analysis of steel structures, Structural Safety 52 (2015) 78–89.

[20] S. M. Vazirizade, S. Nozhati, M. A. Zadeh, Seismic reliability assessment of structures using artificial neural network, Journal of Building Engineering 11 (2017) 230–235.

[21] S. Ghosh, A. Roy, S. Chakraborty, Support vector regression based metamodeling for seismic reliability analysis of structures, Applied Mathematical Modelling 64 (2018) 584–602.

[22] M. A. Hariri-Ardebili, F. Pourkamali-Anaraki, Support vector machine based reliability analysis of concrete dams, Soil Dynamics and Earthquake Engineering 104 (2018) 276–295.

[23] J. Zhang, M. Xiao, L. Gao, An active learning reliability method combining kriging constructed with exploration and exploitation of failure region and subset simulation, Reliability Engineering & System Safety 188 (2019) 90–102.

[24] W. J. de Santana Gomes, Structural reliability analysis using adaptive artificial neural networks, ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg 5 (4).

[25] Z. Jing, J. Chen, X. Li, Rbf-ga: An adaptive radial basis function metamodeling with genetic algorithm for structural reliability analysis, Reliability Engineering & System Safety 189 (2019) 42–57.

[26] M. A. Mendoza-Lugo, D. J. Delgado-Hernández, O. Morales-Nápoles, Reliability analysis of reinforced concrete vehicle bridges columns using non-parametric bayesian networks, Engineering Structures 188 (2019) 178–187.

[27] A. Roy, R. Manna, S. Chakraborty, Support vector regression based metamodeling for structural reliability analysis, Probabilistic Engineering Mechanics 55 (2019) 78–89.

[28] Z.-H. Zhou, Ensemble learning, in: Machine learning, Springer, 2021, pp. 181–210.

[29] Z. Xu, J. H. Saleh, Machine learning for reliability engineering and safety applications: Review of current status and future opportunities, Reliability Engineering & System Safety 211 (2021) 107530.

[30] D. Gómez, A. Rojas, An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification, Neural Computation 28 (1) (2016) 216–228.

[31] A. Subasi, Practical Machine Learning for Data Analysis Using Python, Academic Press, 2020.

[32] D. A. Pisner, D. M. Schnyer, Support vector machine, in: Machine learning, Elsevier, 2020, pp. 101–121.

[33] C.-C. Hsu, Y.-P. Huang, K.-W. Chang, Extended naive bayes classifier for mixed data, Expert Systems with Applications 35 (3) (2008) 1080–1083.

[34] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, R. Strachan, Hybrid decision tree and naïve bayes classifiers for multi-class classification tasks, Expert systems with applications 41 (4) (2014) 1937–1946.

[35] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, Vol. 2, Springer, 2009.

[36] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[37] J. B. Cardoso, J. R. de Almeida, J. M. Dias, P. G. Coelho, Structural reliability analysis using monte carlo simulation and neural networks, Advances in Engineering Software 39 (6) (2008) 505–513.

[38] W. Du, Y. Luo, Y. Wang, Time-variant reliability analysis using the parallel subset simulation, Reliability Engineering & System Safety 182 (2019) 250–257.

[39] H. Haario, E. Saksman, J. Tamminen, Componentwise adaptation for high dimensional mcmc, Computational Statistics 20 (2) (2005) 265–273.

[40] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al., Array programming with numpy, Nature 585 (7825) (2020) 357–362.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.

[42] F. Chollet, et al., Keras: The python deep learning library, Astrophysics source code library (2018) 1806.

[43] F. McKenna, Opensees: a framework for earthquake engineering simulation, Computing in Science & Engineering 13 (4) (2011) 58–66.

[44] P. Raychowdhury, T. C. Hutchinson, Shallowfoundationgen opensees documentation, Open System for Earthquake Engineering Simulation (OpenSEES): University of California, San Diego.

[45] B. D. Scott, R. Park, M. J. Priestley, Stress-strain behavior of concrete confined by overlapping hoops at low and high strain rates, in: Journal Proceedings, Vol. 79, 1982, pp. 13–27.

[46] L. Wang, J. Liu, C. Yang, D. Wu, A novel interval dynamic reliability computation approach for the risk evaluation of vibration active control systems based on pid controllers, Applied Mathematical Modelling 92 (2021) 422–446.

[47] L. Wang, Y. Liu, M. Li, Time-dependent reliability-based optimization for structural-topological configuration design under convex-bounded uncertain modeling, Reliability Engineering & System Safety 221 (2022) 108361.

[48] L. Wang, Y. Liu, D. Liu, Z. Wu, A novel dynamic reliability-based topology optimization (drbto) framework for continuum structures via interval-process collocation and the first-passage theories, Computer Methods in Applied Mechanics and Engineering 386 (2021) 114107.