

Probabilistic Method for Time-Varying Reliability Analysis of Structure via Variational Bayesian Neural Network

Hung V. Dang^{a,b,*}, Ramona Trestian^a, T. Bui-Tien^c and Huan X. Nguyen^a

^aLondon Digital Twin Research Centre, Faculty of Science and Technology, Middlesex University, London

^bFaculty of Building and Industrial Construction, National University of Civil Engineering, Hanoi, Vietnam

^cDepartment of Bridge and Tunnel Engineering, Faculty of Civil Engineering, University of Transport and Communications, Hanoi, Vietnam

ARTICLE INFO

Keywords:

structural engineering
machine learning algorithms
reliability
stochastic processes
numerical simulation.

ABSTRACT

This study proposes a novel framework computing the dynamic reliability and associated uncertainty quantification of structures under time-varying excitation with significantly reduced time complexity. For this purpose, the deep neural network's power and the Bayesian theory's probabilistic ability are leveraged, forming a Bayesian neural network data-driven model (BNN). The BNN-based surrogate model can yield a probability distribution of outputs of interest, e.g., a limit state function and its derived statistics such as median value, confidence interval rather than only a deterministic quantity. The effectiveness and correctness of the proposed method are reaffirmed via three case studies involving examples from the literature and a 3D numerical model of a prestressed reinforced concrete bridge structure, showing a reduction in time complexity up to three orders of magnitude compared to the Monte Carlo method only using finite element models. As a result, an 11-year maintenance routine is recommended for a marine and chemically aggressive environment to ensure the high reliability of prestressed bridge structures when accounting for uncertainty estimation.

1. Introduction

The digital twin is one of the most currently exciting technologies which aims to build a digital model evolving in a synchronized fashion with physical assets, predicting their performance, remaining useful life, and preventing critical failures. When developing a digital twin framework for civil infrastructures, it is unavoidable to assess impacts of the inherent uncertainty of input variables such as material properties, external excitation, and noisy measured data on structural performance, usually by performing reliability analysis. Furthermore, the probabilistic properties of these variables could vary over time; thus, static reliability analysis cannot reflect the time evolution of quantities of interest. That is why the dynamic reliability analysis gains increasing attention and is also the main focus of this study.

In the following, works in the literature related to time-varying structural reliability analysis, which provide the theoretical foundations and motivation for this work, are briefly reviewed. Jiang et al. [17] explored a clear and practical outcrossing rate method to investigate the dynamic reliability of structures with material degradation in time, and reported that the method could accurately reflect the decrease of reliability index under degradation with a relative error smaller than 5%. Fang et al. [13] presented a novel equivalent most probable point addressing time-variant reliability constraints in structural design optimization. The results showed a significant reduction in the number of time-variant reliability analyses required to obtain optimal solutions through two structural examples involving a simply supported beam and a welded beam under dynamic loadings. In [16], reliability and risk assessment of 2D steel frame structures in re-

sponse to time-dependent loads caused by earthquakes were explored using a stochastic approach called SFEM, combining the numerical finite element method with FORM/SORM-based reliability analysis. It turned out that the proposed algorithm effectively quantified the risk of the system in the presence of different uncertainty sources and also provided evidence of improved behavior of the steel frame under earthquake with the presence of reinforced concrete shear walls.

Typically, the direct Monte-Carlo simulation (MCS), with its mathematical clarity and straightforward realization, is able to yield satisfying results for small and simple structures but requires significant computational effort or even becomes practically intractable when the complexity of the structure increases. Fan and Liu [12] developed an effective Gaussian mixed particle filter method using daily extreme data to predict the structural reliability of the North-bound steel bridge in the USA. The obtained results had a reasonable variation, which was smaller than the uncertainty of monitored values, thus promoting the bridge's proactive maintenance. In order to construct the time evolution of the reliability of a deteriorating structure without repeating the whole computational procedure every time instant, Straub et al. [35] combined sampling-based methods for identifying the interval failure probabilities with the first-order reliability approximation of the system. Compared with the direct Monte Carlo, a 10-fold reduction in the number of evaluations and the computational time were observed. Kavel et al. [19] presented a compelling method combining the Asymptotic Sampling technique able to magnify the standard deviation of random variable distributions with the Weighted Simulation technique able to reduce the number of limit state function evaluations. The hybrid method was proven to provide rare failure probabilities with high accuracy and several times reduction in computational efforts compared to coun-

*Corresponding author

 hungdv@nuce.edu.vn (Hung V. Dang)

ORCID(S): 0000-0003-3384-6427 (Hung V. Dang)

terparts such as MC and conventional Asymptotic Sampling. In [14], Han et al. demonstrated the validity of a mathematical model, namely cellular automata, for time-varying reliability analysis of steel structures under corrosion due to the marine environment. Moreover, the method could be extended to various corrosion-related damages such as thickness loss, pit nucleation, and pit propagation; but calibration was needed for different exposure conditions and specific structures. Another effective and widely adopted family of methods [21, 22] is to leverage optimization algorithms to reduce the variable space complexity, thus decreasing the time as resource footprints of the reliability analysis while still maintaining equivalent results compared to Monte Carlo simulation. Despite being accurate, these physical and/or mathematical-based methods require complex limit state theories, which is problematic for large structures as the number of failure modes increases exponentially with the number of sub-components. Moreover, in both cases, no uncertainty estimation is given, which could have an important impact on the safety assessment of real structures [44].

Recently, a novelty surrogate model class based on the data-driven method for the reliability analysis has gained increasing attention among the engineering community thanks to its strong capacity in learning nonlinear relationships between structural response and the high dimensional random-variate input space with reduced time complexity. One of the pioneering works using AI algorithms as a surrogate model to approximate the limit state function is the work of Cheng et al. [7]. Through comprehensive examples, the authors demonstrated the viability and working flows of the approach, providing ideas for subsequent works [20] dealing with more complicated structures using more complex AI architectures. Kalatjari et al. [18] proposed an artificial intelligence-based strategy to quickly find failure paths when performing the reliability analysis of highly redundant structures such as truss structures. The approach's effectiveness was demonstrated to several trusses, including a large one with up to 120 members. Xiang et al. [41] merged deep neural network (DNN) and weighted sampling into one active learning framework for structural reliability assessment of dynamic systems. The DNN was adopted as a surrogate model to predict failure probability instead of using a limit state function, and the weighted sampling was used to update training data for DNN by focusing on the important region close to the limit state surface. Xiao and Lu [42] carried out a structural reliability sensitivity analysis to explore the influence of input variables as well as their interaction effect on structures' failure. The authors mainly used conditional distributions of input variables and the failure-conditional PDF to form a new set of reliability sensitivity indices without requiring any physical model. Su et al. [37] approximated nonlinear performance functions in traditional reliability methods by using a Gaussian Process Regression model. When applying to complex structures such as a truss-type girder and a large suspension bridge, the proposed method was shown to deliver validated insights into the reliability of the structure at a significantly reduced computational cost. However, conventional deep-

learning networks usually require a large amount of training data to perform reliably; otherwise, insufficient data leads to inaccurate results, i.e., overfitting problems. Moreover, in both cases, no uncertainty estimation is given.

In an attempt to address the dynamic reliability of complex structures under time-varying excitation or/and component degradation, this work explores a probabilistic framework based on the Bayesian Deep Neural Network and Variational Inference (VI) inspired from the idea in [23]. The framework leverages the advances of Deep learning, Bayesian probability theory, and probabilistic programming to provide probability distribution of the quantities of interest rather than point estimates even with limited data. Furthermore, with more available data, assessments are improved, and uncertainty is narrowed down. The main contribution of this paper is three-fold:

- The present framework is able to provide the evolution of reliability over time of complex structures, as well as confidence interval (CI) with reduced time complexity (up to three orders of magnitude) compared to finite element model-based methods.
- The framework is flexible and can be adapted for different types of structures, such as high-rise buildings or long-span bridges, as demonstrated through three case studies with increasing levels of complexity.
- Influences of different degradation scenarios on the global reliability of structures are quantified, providing more evidence for engineers in optimizing maintenance strategies. As a result, an 11-year maintenance routine is recommended for a marine and chemically aggressive environment to ensure the high reliability of prestressed bridge structures.

The remainder of this paper is organized as follows. Section 2 presents the mathematical foundations of BNN and VI, as well as its pseudo algorithm. Section 3 describes three study cases involving a mathematical benchmark problem, a 2D multi-story frame, and a 3D prestressed reinforced concrete bridge structure. Finally, Section 4 draws the conclusion and gives some ideas for future work.

2. Probabilistic Framework for Dynamic Structural Reliability Analysis

The workflow of the present approach is diagrammed in Fig. 1, consisting of three successor stages. The first stage involves time-invariant random variable X and time-variant random process $Y(t)$ inputs described by their distribution types and predefined statistical parameters. Specifically, in this work, X are geometric parameters of the structure and random external excitation, while $Y(t)$ reflects the degradation in material properties. Next, the Variational BNN model is designed and trained, whose input data are random variables from stage 1. The outputs of interest are values of the limit state functions of the structure obtained either

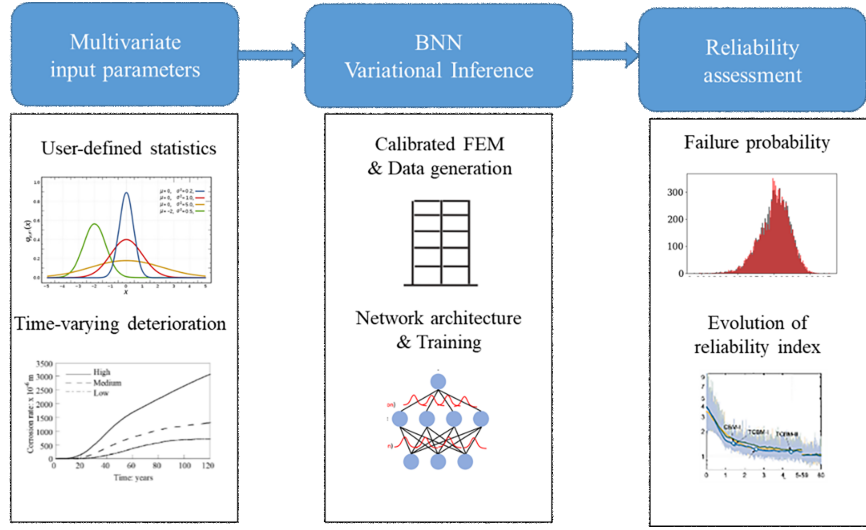


Figure 1: Working flow of the BNN-based reliability analysis method.

by theoretical limit state functions or numerical simulation. Finally, the resulting BNN model is used to assess the evolution of the structure's reliability over time. Additionally, with advantages in reducing time inference complexity and performing probabilistic prediction, parametric studies can be carried out to quantify the impacts of different degradation scenarios on the operation of the structure. In the following paragraphs, details of each stage are described and clarified through case studies further.

2.1. Dynamic Structural Reliability Analysis

Considering a structure with N inputs $X = (x_1, \dots, x_N)$, where x_i is a random variable denoting uncertainty in geometry, loads, or material properties with a prescribed distribution such as normal distribution, Gumbel distribution, etc. A limit state of the structure is expressed via a function $g(X)$ such that $g(X) > 0$ corresponding to a safety state, otherwise, the failure happens, namely $g(X) \leq 0$. Then the probability of the failure P_f is calculated as follows [36]:

$$P_f = Pr[g(X) \leq 0] = \int \dots \int_{g(X) \leq 0} \rho(X) dX, \quad (1)$$

where $\rho(X)$ is the joint probability density function of X . Next, the reliability of the structure, which is the complement of the failure, is given by:

$$R = 1 - P_f = Pr[g(X) > 0], \quad (2)$$

and the reliability index β is obtained via the following equation:

$$\beta = -\Phi^{-1}(P_f), \quad (3)$$

where Φ^{-1} represents the inverse cumulative distribution function of a standard normal random variable. The integration in Eq. (1) can be approached via a plain Monte Carlo simulation directly calculating the limit state function with a large

number of samples of random variables drawn from predefined distributions. Let's introduce I a binary function indicating the safety of a structure as below:

$$I[g(X)] = \begin{cases} 0 & \text{if } g(X) > 0 \text{ (safe)} \\ 1 & \text{if } g(X) \leq 0 \text{ (failure)} \end{cases} \quad (4)$$

The probability of failure is obtained as an expectation of the indicator function over the total number of samples N_s as follows:

$$P_f = \frac{1}{N_s} \sum_{j=1}^{N_s} I[g(X_j)] \quad (5)$$

When the distribution of random variable modifies over time, such as material degradation, we interest in estimating how the structure's reliability evolves correspondingly, as shown in Fig. 2. The time-varying random variable is mathematically described by a stochastic process with its expected value and autocorrelation function. A realization of a stochastic process consists of a set of random variables $Y(t)$ with $t \in [0, T]$ denotes time. Then, the first passage failure scheme is adopted to assess the structure's operational state. Hence, time-varying failure probability is calculated by:

$$P_f(t) = Pr[g(X, Y(t), t) \leq 0 \quad \forall t \in [0, T], \quad (6)$$

$P_f(t)$ can be interpreted as the cumulative probability of failure. If the time discretization is applied, Eq. (6) is rewritten as follows:

$$\begin{aligned} P_f(0, t) &= P\{\cup_{j=1}^r g(X, Y(t_j), t_j) \leq 0, t_j \in [0, t]\} \\ &= P\{\max_{j=0,1,\dots,r} g(X, Y(t_j), t_j) \leq 0, t_j \in [0, t]\}. \end{aligned} \quad (7)$$

Then, the reliability at time instant t reflecting the probability of no failure occurring before t , is calculated as below:

$$R(t) = 1 - P_f(t), \quad (8)$$

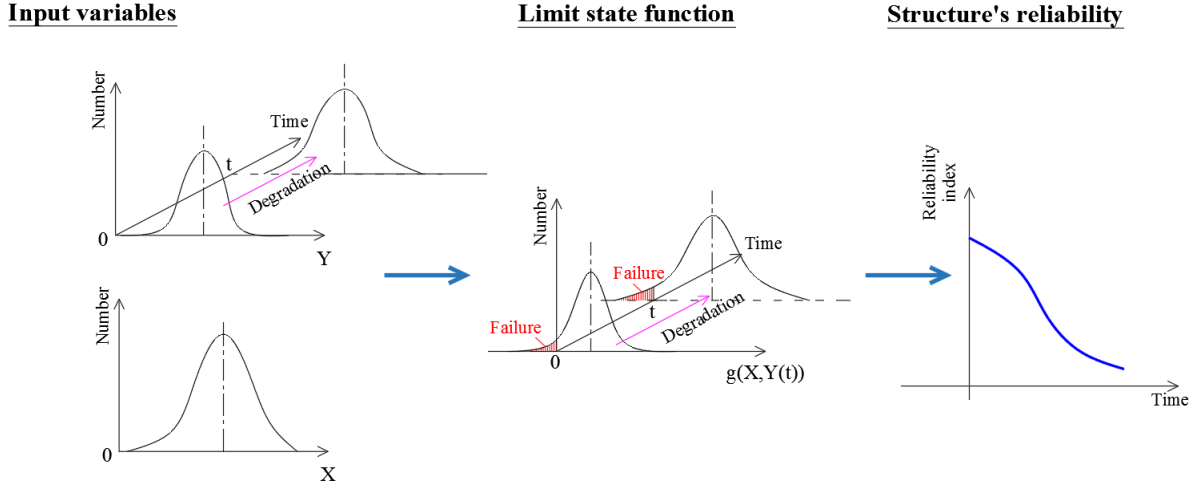


Figure 2: Time varying structural reliability analysis

In order to perform the MCS, the time interval is firstly discretized into N_t instances $[0, t_1, t_2, \dots, t_{N_t} = T]$ usually with a constant time step dt . A sample realization of $Y(t)$ consists of a sequence of values $[y_0, y_1, \dots, y_{N_t}]$, with y_i corresponding to instant t_i . For drawing these sequences of values, it is necessary to specify the type of distribution and correlation functions associated with $Y(t)$ before using the Cholesky decomposition algorithm to generate samples [38, 26]. After that, for each trajectory, the limit state function is evaluated at every time instant until a failure occurring or the final time instant T is reached.

Logically, the computation complexity of time-varying reliability analysis grows rapidly with increasing time intervals and structures' size. Therefore, to make this kind of analysis feasible and practical, two major classes of methods are commonly utilized: i) reducing the number of direct simulations, ii) developing surrogate models more efficiently than direct simulation counterparts. The next section will present a BNN-based surrogate model, which encompasses the power of deep neural networks and the probabilistic ability of Bayesian theory to predict the limit state function conditional on a given set of training data obtained from direct MC simulation.

2.2. Multi-layer Perceptron Network

Conceptually, a deep neural network mimics the information analysis process of the brain, in which meaningful information is extracted from data through a complex system of neurons jointly connected. The most common neural network has a plain architecture composed of multiple layers: input, hidden, and output; each comprises multiple perceptrons. Together they form a multi-layer perceptron network (MLP) mapping input data to output results. Mathematically, one can formulate this nonlinear mapping as follows [6]:

$$Z = F(X, Y|\mathbf{W}) = f_L(\dots f_2(f_1(X, Y|W_1)|W_2)W_L), \quad (9)$$

where L is the total number of layers in the network, $\mathbf{W}$ is the matrix of the network's parameters, W_i is the vector of parameters at layer i corresponding to the row i of $\mathbf{W}$, X, Y and Z denote the time-invariant and time-variant inputs and output vectors, respectively. The connection between two consecutive layers is performed by combining a linear matrix operation and a nonlinear activation function as follows:

$$f_l(X_l|W_l) = h(W_l \otimes [X_l, Y_l] + b_l) \quad \text{with } l = 1, 2, \dots, L, \quad (10)$$

where $\otimes$ is the element-wise product, $[X_l, Y_l]$ and b_l are the inputs and bias vector of the layer l , respectively, and h denotes the nonlinear activation function. In this study, one adopts the ReLU function for hidden layers. The value of network parameters are randomly initialized, then they are iteratively updated via the backpropagation algorithm to minimize a loss function L , reflecting the deviation prediction values and true values. In this study, the root mean square error (RMS) function is adopted for L . The training process terminates when a tolerable error level is met or the number of iterations reaches a limit. It is noted that input variables X, Y , and output Z have different ranges of values; hence, they are standardized to zero mean and unit variance by using their statistical characteristics to eliminate the scale difference between variables. Finally, the resulting output of the model will be rescaled to the original range.

2.3. Variational Bayesian Neural Network

One of the biggest downsides of the Deep Learning model is the requirement of a large volume of curated data, which is usually not the case in practice. Thus, [5] proposed to incorporate the probabilistic theory into the Deep Learning model to address this problem. By doing so, network parameters $\mathbf{W}$ of a probabilistic neural network are characterized by their distributions instead of an optimal set of values, the prior distribution of $\mathbf{W}$ can be initialized with one of the popular distributions such as uniform or Gaussian ones, as shown

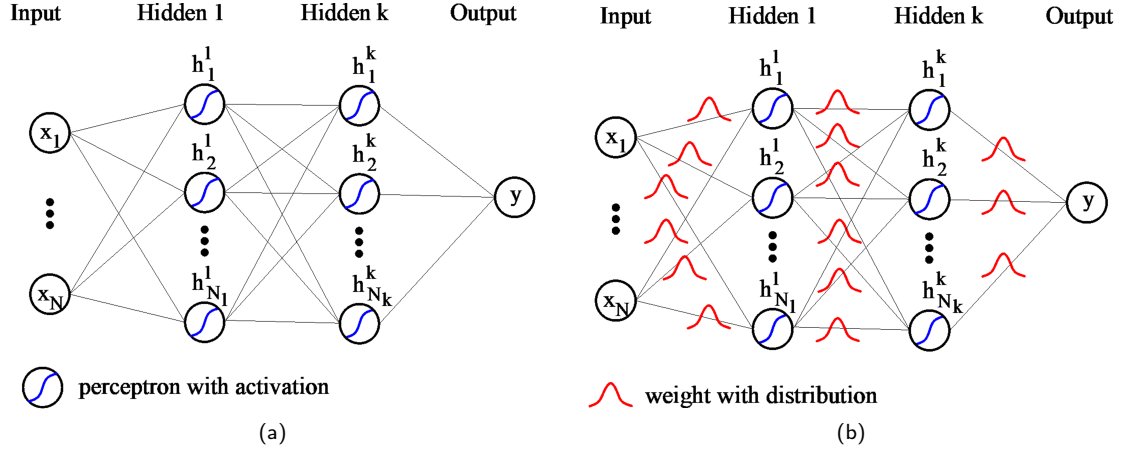


Figure 3: Typical architectures of (a) conventional Multiple Layer Perceptron and (b) Bayesian Neural Network.

in Fig. 3. While the posterior distribution $p(\mathbf{W}|D)$ given a dataset $D = (X, Y, Z)$ will be determined through a training process. At the inference stage, the distribution of output values $p(Z|D, \mathbf{W})$ is derived au lieu of a point estimate. However, exact solutions for the posterior $p(\mathbf{W}|D)$ are practically untractable due to complicated integral operations. Thus, Variational Inference is resorted to [24]. The true posterior $p(\mathbf{W}|D)$ is approximated by a family of variational distributions $q(\mathbf{W}|\theta)$ parameterized by θ . The training process is to seek for θ^* to minimize the discrepancy between true distribution $p(\mathbf{W}|D)$ and approximate one $q(\mathbf{W}|\theta)$. The metric for measuring this discrepancy is the Kullback-Leibler divergence (KL) [25] expressed as follows:

$$KL(q(\mathbf{W}|\theta)||p(\mathbf{W}|D)) = E_{q(\mathbf{W}|\theta)} \log \frac{q(\mathbf{W}|\theta)}{p(\mathbf{W}|D)}. \quad (11)$$

Then, θ^* is the solution of the following minimization problem:

$$\theta^* = \arg \min_{\theta} KL(q(\mathbf{W}|\theta)||p(\mathbf{W}|D)). \quad (12)$$

Via the Bayes' rule, $p(w|D)$ is expressed as:

$$p(\mathbf{W}|D) = \frac{p(D|\mathbf{W})p(\mathbf{W})}{p(D)}. \quad (13)$$

After mathematical transformations, (11) is rewritten as

$$KL[q(\mathbf{W}|\theta)||p(\mathbf{W}|D)] = L(\mathbf{W}, \theta) + \log p(D), \quad (14)$$

where $L(\mathbf{W}, \theta)$ is known as evidence lower bound loss (ELBO) [30]:

$$L(\mathbf{W}, \theta) = E_{q(\mathbf{W}|\theta)} \log q(\mathbf{W}|\theta) - E_{q(\mathbf{W}|\theta)} \log p(\mathbf{W}) - E_{q(\mathbf{W}|\theta)} \log p(D|\mathbf{W}). \quad (15)$$

Because $p(D)$ does not depend upon θ , minimizing KL divergence is equivalent to minimizing the above loss function

$L(\mathbf{W}, \theta)$, which is numerically approximated by a sampling method as follows:

$$L(\mathbf{W}, \theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} [\log q(\mathbf{W}^i|\theta) - \log p(\mathbf{W}^i) - \log p(D|\mathbf{W}^i)], \quad (16)$$

where N_s is the total number of samples.

Besides, one adopts uncorrelated multivariate Gaussian distribution for variational distributions $q(\mathbf{W}|\theta)$, i.e., $\theta = (\boldsymbol{\mu}, \boldsymbol{\sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ stand for matrices of mean and standard deviation, respectively, because of its practicality and popularity [2, 43]. The mathematical expression of weight matrix is rewritten as below:

$$\mathbf{W} = \boldsymbol{\mu} + \boldsymbol{\sigma} \times \boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \quad (17)$$

On the other hand, it is expected that the training time of BNN models will be two times longer than a frequentist network with the same architecture because each weight of BNN requires two parameters, i.e., its mean and variation instead of only one estimated value. The gradient of the loss function with respect to the mean $\boldsymbol{\mu}$ is given by:

$$\Delta \boldsymbol{\mu} = \frac{\partial L(\mathbf{W}, \theta)}{\partial \mathbf{W}} + \frac{\partial L(\mathbf{W}, \theta)}{\partial \boldsymbol{\mu}}. \quad (18)$$

The gradient with regard to the standard deviation $\boldsymbol{\sigma}$ is expressed as:

$$\Delta \boldsymbol{\sigma} = \frac{\partial L(\mathbf{W}, \theta)}{\partial \mathbf{W}} \times \frac{\boldsymbol{\epsilon}}{\boldsymbol{\sigma}} + \frac{\partial L(\mathbf{W}, \theta)}{\partial \boldsymbol{\sigma}}. \quad (19)$$

The variational parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ of $\mathbf{W}$ are updated as follows:

$$\begin{aligned} \boldsymbol{\mu} &\leftarrow \boldsymbol{\mu} - \alpha \times \Delta \boldsymbol{\mu}, \\ \boldsymbol{\sigma} &\leftarrow \boldsymbol{\sigma} - \alpha \times \Delta \boldsymbol{\sigma}. \end{aligned} \quad (20)$$

```

1  begin
2
3  Data:
4  Discretize the time interval  $[0; T]$  with a time step  $dt$ .
5  Generate time-invariant and variant random variables.
6  Function Training BNN network
7      Input:
8          Structural random variables, e.g. Tables 3 and 5.
9          Limit state function values using predefined functions, for example, Eqs. (21), (24) and (29).
10     Initialize BNN
11         Seek for an appropriate frequentist MLP architecture via the Grid search
12         Assign normal distributions for MLP weights as prior probabilities per Eq. (17)
13         Define a family of variational distribution to approximate the posterior probability
14     for  $i \leftarrow 1$  to Number of epochs do
15         Compute outputs and associated uncertainties by performing multiple predictions with network's weights
16         drawn from  $q(\mathbf{W}|\theta)$ 
17         Calculate the loss function ELBO according to Eq. (16)
18         Update variational parameters of the network's weights by Eq. (20)
19     end
20     Output:
21         Posterior weight distributions  $\mathcal{N}(\mu, \sigma)$ 
22 End Function
23 Function Dynamic reliability analysis
24     Input:
25         Generate  $N_{traj}$  trajectories of random processes from 0 to  $T$ .
26         using time-varying statistical characteristics as described by Eqs. (22), (23) and (26).
27     for  $i \leftarrow 1$  to  $N_{traj}$  do
28         | Estimate limit state function values and their 90% CI using the trained BNN model
29     end
30     for  $t \leftarrow 0$  to  $T$  do
31         | Calculate probability failure and reliability index at  $t$  by using Eqs. (5) and (3)
32     end
33 End Function
34 Result: Time evolution of reliability index and its confidence interval.
35 end
    
```

Algorithm 1: Variational BNN algorithm for time-varying structural reliability analysis

2.4. BNN-based Analysis Procedure

Having introduced the theoretical foundation of the time-varying reliability analysis and BNN, Algorithm 1 presents the main realization steps of the proposed BNN framework. The implementation of the BNN is realized with the help of the probabilistic library Pyro [4] written in Python and supported by the machine learning library Pytorch, thanks to its expressiveness, flexibility, and robustness acknowledged in the literature [31]. In order to achieve high performance, some additional steps are conducted before those mentioned in Algorithm 1 such as data standardization to suppress the scale difference of input variables, grid search to determine an appropriate frequentist MLP architecture.

3. Case study

In this section, the BNN-based method for time-varying reliability assessment is applied to three case studies ranging from a relatively simple math problem to a 2D steel frame and a 3D complex bridge structure. The two first examples are selected due to their availability in the literature, thus allowing comparing the validity of the proposed method with existing counterparts, whereas the third one aims to demonstrate its effectiveness for a real-world application. For all

case studies, the same high-level workflow described previously is carried out, though some hyperparameters of the BNN models are problem-specific, which will be explicitly mentioned later.

3.1. Case Study 1: Mathematical example

Table 1
Random variables for the mathematical example

Variable	X_1	X_2	$Y(t)$	t	$g(X, Y)$
Distribution	Normal	Normal	Normal		
Mean	3.5	3.5	0		
Std	0.25	0.25	1		
Min	2.49	2.45	-4.12	0	-79.23
Max	4.60	4.46	4.81	1	66.79

The mean and standard deviation of g vary over time t .

The first case study is a mathematical example came from [40] considering a time-varying limit state function $g(X, Y(t), t)$ as follows:

$$g(X, Y(t), t) = 20 - X_1^2 X_2 + 5X_1(1 + Y(t))t - (X_2 + 1)t^2, \quad (21)$$

where $X = [X_1, X_2]$ are time-independent random variables following a normal distribution with a mean of 3.5 and a

standard deviation of 0.25, t is time variable varying in the range $[0, 1]$, and $Y(t)$ is a zero-mean unit variance Gaussian Process following a squared exponential kernel variance function as below:

$$\rho(t_1, t_2) = \exp(-(t_2 - t_1)^2). \quad (22)$$

At first, a Monte Carlo simulation of 10^4 samples is realized to create a database for training BNN models. The database has a tabular form composed of four columns for input values $X_1, X_2, t, Y(t)$, and output value $g(X, Y(t), t)$. Next, the database is split into non-overlapping training and validation datasets with a ratio of 80:20. Afterward, a deterministic MLP model is built to predict the limit state function, of which the adequate architecture is determined by a preliminary grid search considering different combinations of numbers of hidden layers and numbers of perceptrons in each hidden layer. Further details of the grid search can be found in [3]. The preliminary grid search conducted by the authors results in a MLP architecture of 4/8/8/1, meaning an input, two hidden, and output layers with 4, 8, 8, and 1 perceptron, respectively. Having defined the network architecture and prepared the database, the training process is carried out to determine parameters of BNN's weight distributions, namely, mean and standard deviation values.

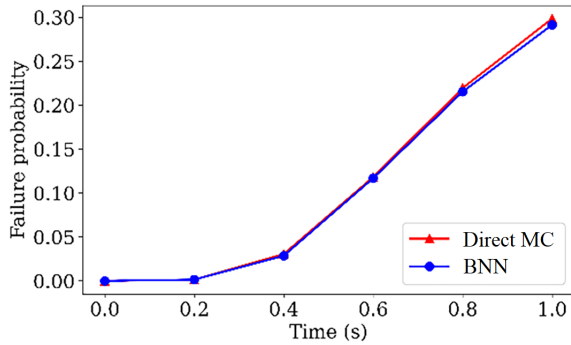


Figure 4: Comparison of failure probabilities obtained by BNN with those reported in [40]

Table 2

Comparison of reliability analysis results obtained by BNN with those reported in [40]

Time (s)	Direct MC		BNN		$\Delta\beta$ (%)
	P_f (%)	β	P_f (%)	β	
0.2	0.00144	2.98	0.0015	2.97	0.4
0.4	0.03050	1.87	0.0286	1.90	1.7
0.6	0.11820	1.18	0.1169	1.19	0.9
0.8	0.21952	0.77	0.2153	0.79	2.3
1.0	0.29860	0.53	0.2914	0.55	3.6

Total number of samples: 10^6

CPU time (s): 0.77 s for Direct MC and 4.6 s for BNN

Once the BNN model is trained, it is used to estimate

the limit state function and then derive the system's reliability. The obtained results will be subsequently compared with the direct MC using Eq. (21) in terms of probability failure, reliability index as well as CPU time. Fig. 4 plots the failure probability as a function of times using the BNN model and results reported in [40], showing that two curves are well overlapped, meaning good agreements between results. Additionally, Table 2 details the reliability index β for different time intervals and the deviation between direct MC and BNN-based method. It is clear that the BNN approaches the theoretical solutions with small errors under 5%, which reaffirms the BNN-based approach's correctness. Note that the BNN required only 10^4 samples for the training process but can achieve an accuracy of more than 95% on test data with 10^6 samples. In terms of CPU time, it takes 4.6s for performing 10^6 samples, which is quite larger than that of the MC directly using theoretical solutions, i.e., 0.77s. However, for more complex systems, when the limit state cannot be determined analytically by mathematical equations, the effectiveness of the BNN-based will be more pronounced.

The compelling merit of the BNN method over MLP counterparts is the ability to offer informative results about quantities of interest even with limited training data through a confidence interval rather than an over-confident deterministic estimation. This characteristic is illustrated in Fig. 5, showing the reliability evolution with respect to time by using a BNN model trained on different training datasets whose number of samples varying in the range $[10^3, 10^4, 10^5]$, respectively. Interestingly, the 90% confidence interval (CI) area encloses the ground truth curve generated by the direct MC. Here, the 90% CI provides a range of values containing actual values of the reliability index with a probability of 90%. In other words, the probability of receiving a value outside of this area is no more than 10%. Technically, to obtain these CI, 100 rounds of inference are performed, each round using different weight values sampled from resulting weight distributions of the trained BNN. Note that the reliability index is calculated by Eq. (3). A high positive value of β signifies a small failure probability, e.g., at $t=0$, $\beta=3.0$ corresponds to a $P_f \sim 0.1\%$. In contrast, a low or even negative value of β means a large P_f , for example, at $t=5$ s, $\beta = -1$ corresponds to $P_f = 16\%$. In reality, negative values of β are rarely of interest as real structures need to achieve a reliability threshold prescribed in Building code and Standards. Here, they are reported to provide a whole picture about the evolution of the system over time. In short, this first example helps clarify the steps of realization of BNN-based methods and validates its accuracy, encouraging its application to other case studies.

3.2. Case Study 2: 2D Steel Frame Structure

This section extends the BNN-based method to assess the dynamic reliability of a five-story three-bay steel frame structure [27] subjected to lateral loads, as shown in Fig. 6. The frame has a typical story height of 4.0 m, except for the first story (5.0 m). The bay widths are 8.0 m, 10.0 m, and 8.0 m, respectively. There are in total 21 random variable inputs

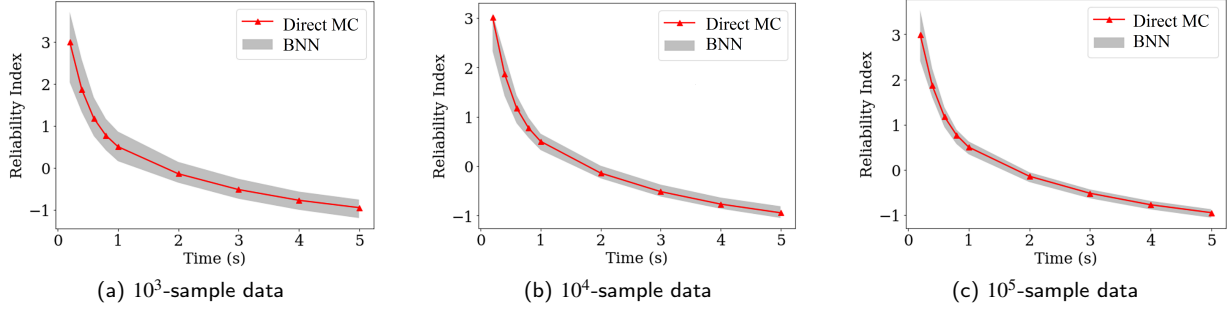


Figure 5: Evolution of reliability index with respect to time for different training data set. BNN trained with 10^3 -sample dataset (a), 10^4 -sample dataset (b), and 10^5 -sample dataset (c), respectively.

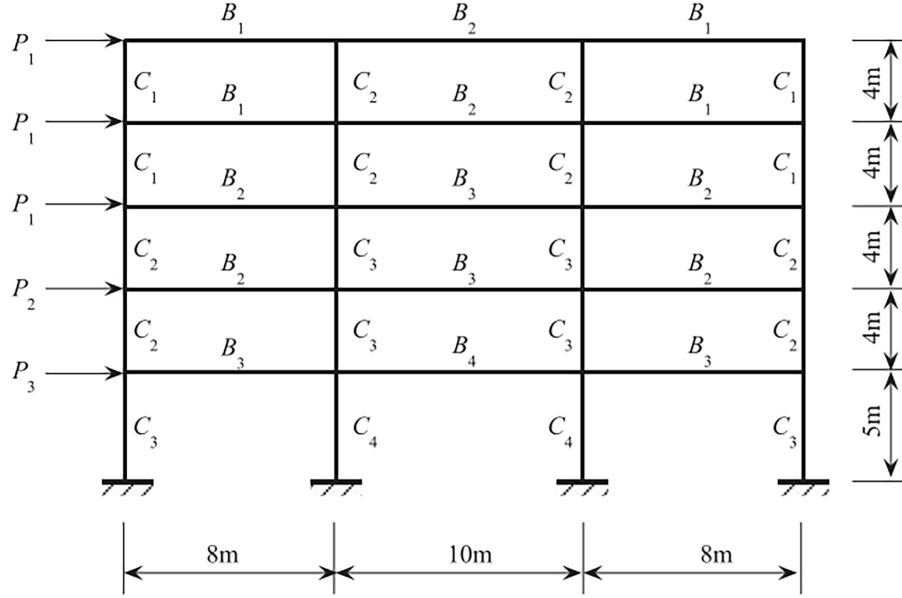


Figure 6: Schematic presentation of five-story three-bay steel frame structure

for this case study, including Young's modulus of columns and beams, geometric properties of members, and horizontal forces, as listed in Table. 3 with corresponding distribution types, mean, and coefficient of variation (*CoV*) values. Particularly, the time-varying material degradation is introduced via an elastic modulus $E(t)$ described as below [28]:

$$E(t) = E_0(1 - \eta_1 \times t^{\eta_2}), \quad (23)$$

where E_0 is the initial value at time $t = 0$, η_1 and η_2 are characteristic parameters taken as $\eta_1 = 0.007$ and $\eta_2 = 0.68$. It is for the sake of brevity to select such a simple and clear equation for modeling material degradation.

The limit state function of the frame is defined by:

$$g(x) = \Delta_{top} - \Delta_0, \quad (24)$$

where Δ_{top} is the horizontal displacement at the top story, Δ_0 is an user-defined threshold. Unlike the first example, there is no existing analytical solution for the limit state function

of such a multi-component structure. Thus, one needs to resort to the finite element method (FEM) to perform the structural analysis, then derive corresponding Δ_{top} . Here, the FEM model of the frame is constructed using the open-source software OpenSees [29] from the Pacific Earthquake Engineering Research Center. The OpenSees is adopted in this work because it is written in the programming language C++; thus, the simulation time is very fast compared to other counterparts, which compensates for its lack of a friendly user graphic interface. Once a reference FEM is established, a database of random variables having 7×10^4 samples corresponding to 7 values of *CoV* is created using predefined probability distribution listed in Table. 3. Next, the database is fed into FEMs to perform MC simulations. The direct MC is accelerated by applying the multiprocessing computation, which allows for performing calculations in parallel on different CPU cores in a computer server. For this example, the computation time takes 59.3 minutes on a server equipped with 12 Intel-Xeon cores, 32 GB RAM. The database, in-

Table 3
Random variables for the five-story three-bay steel frame structure

Variable	$P_1^{(*)}(kN)$	$P_2^{(*)}(kN)$	$P_3^{(*)}(kN)$	$E_1^{(**)}(kN/m^2)$	$E_2^{(**)}(kN/m^2)$	$I_1(m^4)$	$I_2(m^4)$	$I_3(m^4)$
Distribution	Gumbel	Gumbel	Gumbel	Lognormal	Lognormal	Lognormal	Lognormal	Lognormal
CoV	[0.2 ÷ 1.4]	[0.2 ÷ 1.4]	[0.2 ÷ 1.4]	0.05	0.05	0.1	0.1	0.1
Mean	135.0	90.0	70.0	2.1×10^7	2.4×10^7	0.0081	0.011	0.0213
Min	[72.6 ÷ -314.7]	[43.3 ÷ -236.6]	[29.7 ÷ -210.5]	1.70×10^7	1.91×10^7	0.0051	0.0071	0.0133
Max	[371.2 ÷ 1883.7]	[241.0 ÷ 1075.7]	[177.3 ÷ 845.9]	2.68×10^7	2.98×10^7	0.0126	0.0170	0.0327
Variable	$I_4(m^4)$	$I_5(m^4)$	$I_6(m^4)$	$I_7(m^4)$	$I_8(m^4)$	$A_1(m^2)$	$A_2(m^2)$	$A_3(m^2)$
Distribution	Lognormal	Lognormal	Lognormal	Lognormal	Lognormal	Lognormal	Lognormal	Lognormal
CoV	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Mean	0.0295	0.0108	0.0141	0.0232	0.0259	0.312	0.372	0.505
Min	0.0179	0.0070	0.0082	0.0151	0.0170	0.1962	0.2432	0.3310
Max	0.0461	0.0161	0.0220	0.0359	0.0409	0.4763	0.5575	0.7955
Variable	$A_4(m^2)$	$A_5(m^2)$	$A_6(m^2)$	$A_7(m^2)$	$A_8(m^2)$	t (years)	$\Delta_{top}(m)$	
Distribution	Lognormal	Lognormal	Lognormal	Lognormal	Lognormal			
CoV	0.1	0.1	0.1	0.1	0.1			
Mean	0.557	0.253	0.291	0.372	0.418		0.026	
Min	0.354	0.155	0.175	0.236	0.276	0	-0.060	
Max	0.860	0.388	0.455	0.568	0.640	100	0.373	

*The coefficient of variation (Cov) of lateral loads vary from 0.2 to 1.4; hence also do their maximum and minimum values.
 **The mean value of the elastic modulus E_1 and E_2 vary over time according to Eq.(23); hence also do their maximum and minimum values.

cluding 21 random variables and the top story horizontal displacements obtained from FEMs, is then used to train the surrogate BNN model.

Table 4
Comparison results of direct MC and BNN in terms of efficiency

	Training process	Static $t = 0$	Dynamic $t = [0, 100]$
Number of samples	7×10^4	7×10^6	$2 \times 7 \times 10^6$
MC-CPU time (min)	0	512	$\sim 2 \times 512$
BNN-CPU time (min)	59.3	1.67	3.50

In order to identify an appropriate architecture of BNN for this example, a MLP network is first conceived, whose number of hidden layers and number of perceptrons per hidden layer are determined via a grid search [3] using a portion of 10% of the FEM database. As a result, the MLP of 21/64/32/1, having two hidden layers with 64 and 32 perceptrons, respectively, achieved the highest accuracy. After that, the MLP is converted to a BNN by assigning a prior normal distribution with zero mean and unit variance to every weights of the network. The posterior weight distributions' parameters, i.e., mean and variance, are determined via a training process, which minimizes the KL divergence as described above.

Fig. 8 depicts the loss functions, including the mean square error (MSE) and mean absolute value (MAE) obtained for the 21/64/32/1 MLP network on the training and validation

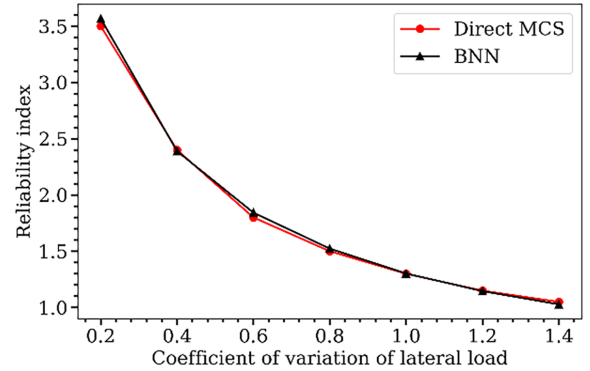


Figure 7: Reliability index in a function of the coefficient of variation of the lateral loads

datasets, showing that the MLP achieved stable results on the training data after 500 epochs, i.e., iterations. However, the validation results fluctuate to some extent. Thus, the probabilistic BNN-based model is resorted to for estimating these uncertainties. Fig. 8(c) displays the Elbo loss function according to Eq. (16) for the BNN model, where stable results are observed for both training and validation datasets.

Next, one approximately estimates the number of calculations when performing reliability analysis by the following equations [10]:

$$N_{sample} = \frac{1 - \hat{P}_f}{CoV^2 \times \hat{P}_f}, \quad (25)$$

where CoV is the coefficient of variance of the computed

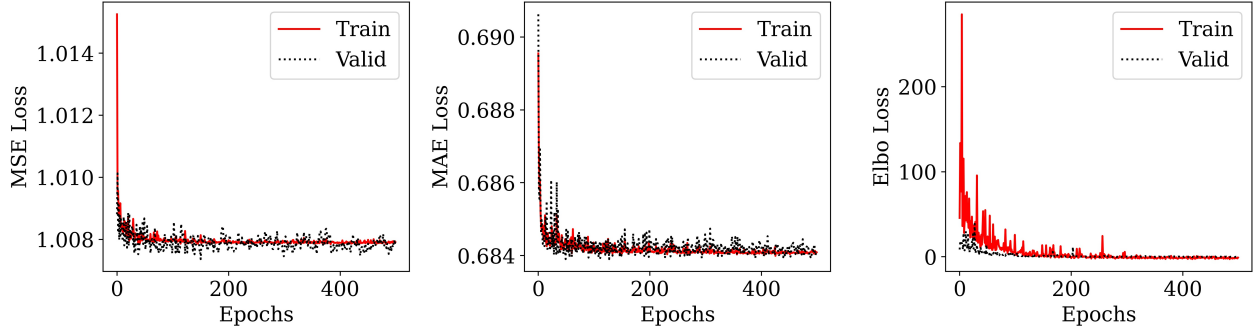


Figure 8: Evolution of loss functions (MAE, MSE and Elbo) for training and validation datasets

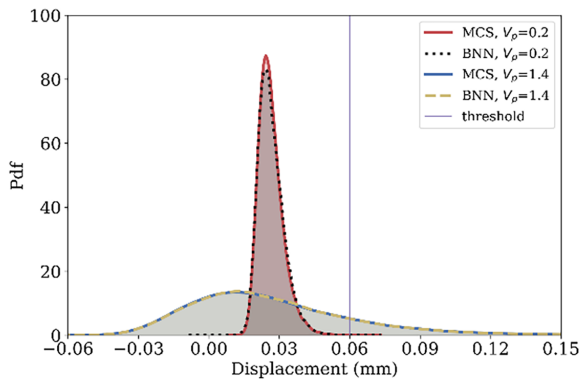


Figure 9: Pdf of top-floor displacement obtained for $V_p = 0.2$ and $V_p = 1.4$ using BNN and direct MCS methods.

results. It can be seen that for a smaller reliability index, the required number of samples is low and vice versa. For example, with a low failure probability $P_f = 0.0001$ equivalent to a high-reliability index of 3.8 and $CoV = 0.1$, the number of sample N_{sample} is around 10^6 .

After that, the obtained BNN is used as a surrogate model to derive the limit state function au lieu of FEM models and perform reliability analysis later. Fig. 7 compares the reliability index at $t = 0$, obtained by BNN with results reported in [27] for different coefficients of variation of lateral loads denoted by V_p from 0.2 to 1.4. Apparently, an excellent agreement is observed as two curves closely overlap each other. In order to obtain each point on the curves, 10^6 samples of calculations were realized, then the top-floor displacements were recorded to compare with the maximum allowable displacement of 0.06 m ($1/350H$). Fig. 9 depicts results for two cases of V_p , i.e., the smallest and largest values (0.2 and 1.4). It can be seen that with higher values of V_p , the probability density function (pdf) of the top-floor displacement spreads over a wider area, thus leading to a larger P_f , i.e., lower β . It is also noticeable that the pdf of BNN is nearly identical to those of MCS, except for a small difference around the mean value. This can be explained by the fact that there are still other parameters contributing to the

behaviors of the structure in FEM in addition to considered random variables. Neglecting these parameters simplified the problem complexity but could lead to some deviations, though minor as observed.

As the performance of the BNN model is confirmed through the comparison with results from the literature, the dynamic-reliability analysis is carried out for a time interval of $[0, 100]$ years over which the material of structural elements degrades gradually. Note that only samples surviving from previous time instants are considered. For simplification, a supplement study carried out by the authors found that a coefficient of 2 between dynamic and static scenarios is fairly enough to provide stable results. In other words, the number of calculations for time-varying reliability analysis is 2×10^6 . It is straightforward to prove that even if the degradation does not occur and a low failure probability of 0.0001 is maintained, such a number of calculations always ensure more than 10^6 samples at each time instant. If degradation happens, failure probability increases, the required number of calculations will be lowered further. Moreover, there are 7 different values of CoV of lateral loads in the range $[0.2, 1.4]$; thus, the total number of calculations for the dynamic case is $2 \times 7 \times 10^6$.

The material degradation is expressed through a decrease of elastic modulus in accordance with Eq. (23) for all the structure's components. Intuitively, the degradation of material properties tends to decrease the structure's capacity; thus, the reliability index β will exhibit a decreasing trend. Fig. 10 quantifies this intuition for different values of V_p in which the solid curves denote their median values while the shaded areas represent a 90% confidence interval (CI). A high value of $\beta = 3.6$ for $V_p = 0.2$ at $t=0$ corresponds to an extremely small failure probability of 0.0001 while for $V_p = 1.4$ at $t=100$ years, the failure probability increases up to 96%, i.e., a negative value of $\beta - 1.8$. In the figure, the CI is obtained by repeating the inferences 100 times with different values of BNN weight parameters drawn from trained posterior distributions. It is argued that there are other ways to generate the CI with a conventional DNN model, possibly by repeating the tedious and time-expensive data generation stages to obtain different data set before performing the in-

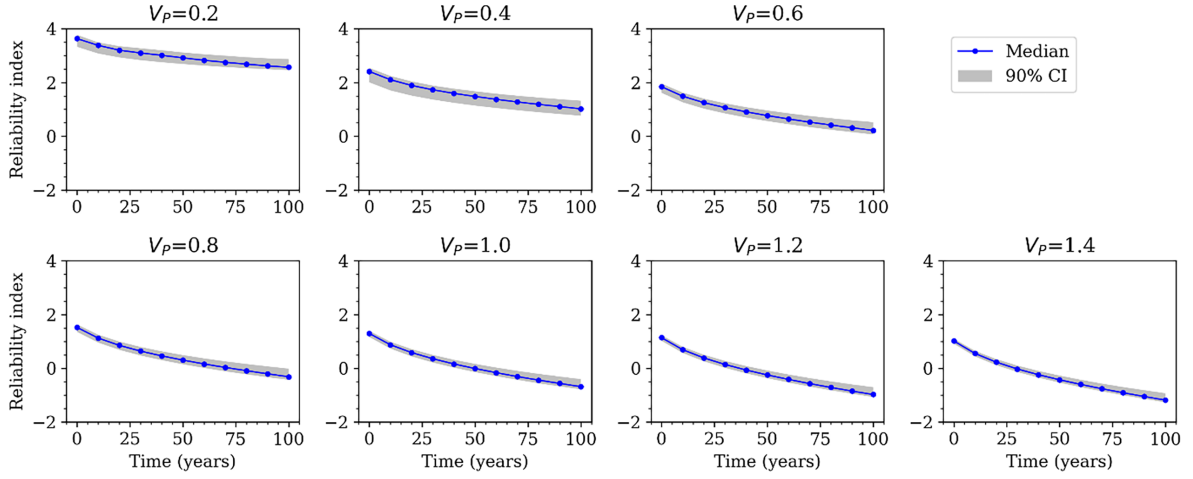


Figure 10: Time-varying of reliability index for different values of V_p from 0.2 to 1.4.

ference. However, a DNN is deterministic in nature, while BNN is probabilistic; hence it enables computing the CI by using only one dataset.

Regarding the time complexity, Table 4 summarizes the CPU time required for each key step in the reliability analysis, clearly showing the superiority of the BNN-based method in reducing time complexity. It takes only 1.67 minutes to accomplish 7×10^6 samples of calculation, whereas that of the direct MCS is 512 minutes, i.e., more than 300-fold reduction. For dynamic reliability analysis over a large time interval $[0, 100]$ years, the CPU time is only around 3.5 minutes. Note that the BNN method necessitates 54.2 minutes for the training process using 7×10^4 samples generated by the direct MCS, but the gain in time complexity is still significant. Here, all calculations were realized on a workstation PC with specifications of CPU Intel Xeon E5-2696, 32 Gb RAM, GPU GTX 1080Ti.

3.3. Case Study 3: Prestressed Bridge Structure

In this subsection, the proposed method is extended to a real-world structure, namely the Doan Hung bridge in Viet Nam, a 36 years old 5-span reinforced concrete bridge with a length of 120 m and a width of 9.0 m. Being part of the national route, there is a considerable volume of traffic crossing the bridge daily; thus, it is important to ensure its reliability. At first, a detailed FEM of the bridge is constructed based on its construction drawings having geometric dimensions and material properties, shown in Fig. 11. The FEM is realized using the Abaqus Standard [34] thanks to its powerful ability in performing dynamic analysis. In addition, Abaqus is equipped with programming-aid features that allow FEMs to connect with other programming software, thus facilitating the analysis with a large number of inputs, as in the case of Monte Carlo analysis. Specifically, in the FEM, the bridge slab and longitudinal girders are modeled by four noded S4R shell elements, and horizontal beams by B31 beam elements, and the prestressed cables by two-node T3D2 truss elements. Displacement coupling constraints are

enforced between these elements to ensure the deformation compatibility criteria when the structure is deformed under excitation. In addition, the bonding between prestressed tendons and the host concrete is enforced by using the Embedded Layer feature in Abaqus, with options for adjusting tendon position from shell mid surface. The truck is assumed to be in contact with the bridge deck all the time, i.e., no bounding movement occurs either penetration of the truck element into the deck surface. Numerically, the master-slave contact algorithm is utilized to model the interaction between the truck versus the bridge. In terms of boundary conditions, the bridge is fixed at two extreme ends and has four intermediate pinned supports corresponding to bridge piers. The spatial mesh size and time step are respectively set to 0.5m and 1×10^{-3} s, which are small enough to ensure convergence for dynamic analysis in this study. Fig. 11(b) illustrates the numerical model of the bridge in Abaqus. Such a model with 2D shell elements is widely accepted for dynamic analysis [15] with reasonable time complexity rather than a detailed model with 3D solid elements but too computationally expensive. For simplicity, the most common type of live load faced in the bridge structure, i.e., vehicular load, is taken as the only excitation source. More specifically, the vehicular load is a single standard HS20 truck of 240 kN according to the American Association of State Highway and Transportation Officials specification [1].

For a prestressed reinforced concrete structure, the prestressed tendons are one of the most critical parts having a significant impact on the stiffness and strength of the structure. Studying prestress loss involves multiple factors and their high-order interactions such as shrinkage, elastic shortening, creep of concrete, steel relaxation, relative humidity as specified by many researchers and Codes [8]. The main interest of this case study is to investigate the applicability of the proposed framework for estimating the time-varying reliability of a real structure which is still hard to accomplish due to the high cost of calculation even when using a sim-

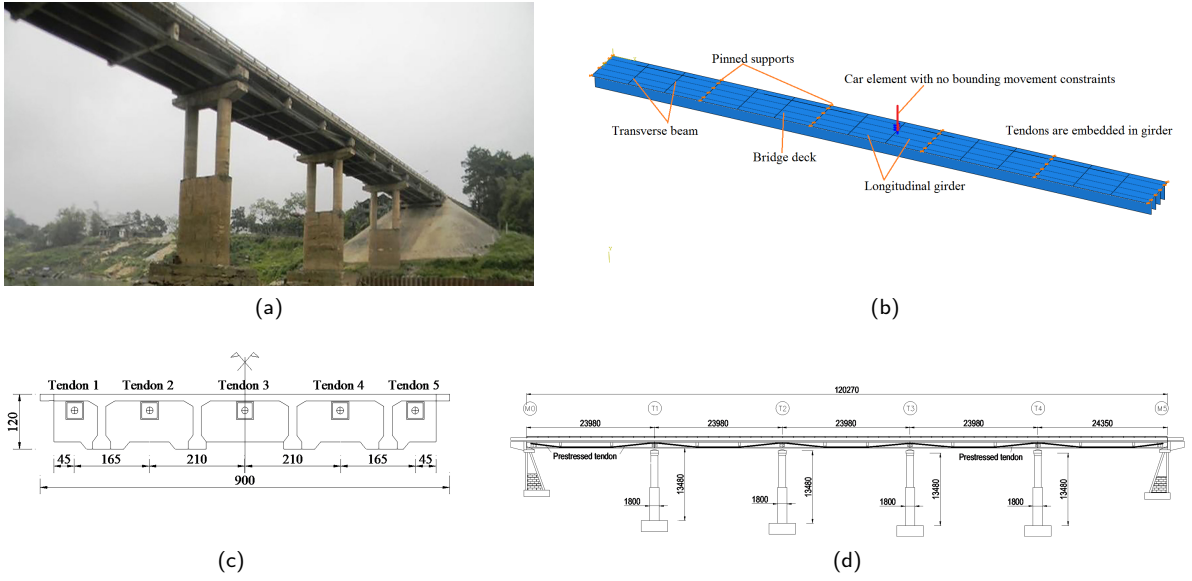


Figure 11: Representation of the prestressed Doan Hung bridge in Viet Nam. a) Image of the bridge. b) Numerical model in Abaqus. c) Lateral section of the bridge. d) Longitudinal section of the bridge

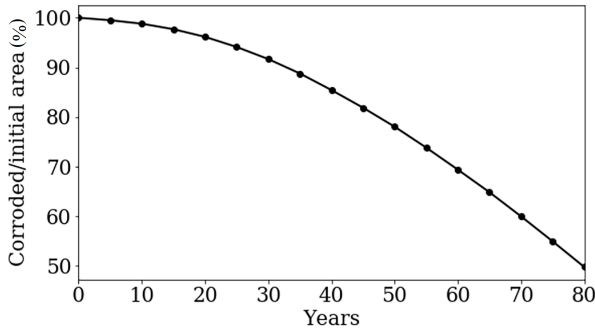


Figure 12: Corrosion of tendon area over time following [39]

plified numerical model. Thus, for simplicity, the losses in prestress forces are indirectly taken into account by considering the area reduction of tendons. In reality, the corrosion of prestressing steel has caused failure in a number of structures such as the Azergues River Bridge in France, the Ynys-y-Gwas Bridge in Great Britain [33], parking garages in the United States, etc. Although only considering area reduction of prestressed steel might under/overestimate the structure's performance, but it provides valuable insight into the effect of such a key factor, thus offering informative arguments for more complex and detailed studies in the future.

One of the practical ways to describe this phenomenon is to assume that corrosion is mainly caused by chloride contamination and is considered to be uniform, as described in the model of Val et al. [39]. Then, the cross-section area reduction is estimated by the following equation and respective chart plotting time history of corroded/initial area ratio

as shown in Fig. 12:

$$D(T) = D_0 - 0.0232 \int_0^T i_{corr}(t) dt, \quad (26)$$

where D_0 is the initial diameter in mm of rebar, i_{corr} denotes the corrosion rate. Suppose a constant annual corrosion rate, the remaining cross-section area is calculated as below:

$$As(t) = \frac{1}{4} \pi [D_0 - 0.0232 \cdot i_{corr} \cdot t]^2. \quad (27)$$

In terms of modeling, each tendon's remained cross-section area is considered one of the random variables following Gumbel distribution with a time-varying mean value according to Fig. 12 and a coefficient of variation of 0.1 as listed in Table 5.

Table 5 enumerates the random variables of interest in this example, including the modulus of elasticity of the material, the thickness of the main structural elements, and the tendon area. The limit state of the structure is determined using the dynamic amplification factor, which is the ratio between the maximum dynamic vertical deflection $y_{max}^{dynamic}$ of the bridge over the maximum static one y_{max}^{static} when considering the truck stops at the middle of the bridge [32], as follows:

$$DAF = \frac{y_{max}^{dynamic}}{y_{max}^{static}}. \quad (28)$$

In order to create data for the training and validation of the BNN model, an extensive series of numerical simulations of the bridge is carried out with the help of Abaqus's Application Programming interface and Python scripts. In this way, it is easy to vary the input parameters and perform analysis in batch mode and in a parallel fashion on different

Table 5
Random variables for Doan Hung bridge's model

Variable	E (kN/m)	Deck thickness (m)	Beam thickness (m)	Girder thickness (m)	Tendon area 1 (mm ²)	Tendon area 2 (mm ²)
Distribution	Lognormal	Lognormal	Lognormal	Lognormal	Gumbel	Gumbel
CoV	0.05	0.05	0.05	0.05	0.1	0.1
Mean	32.5×10^9	0.12	1.2	1.2	2250α	2250α
Min	26.1×10^9	0.096	0.946	0.953	1427α	1417α
Max	40.3×10^9	0.149	1.456	1.481	3504α	3466α
Variable	Tendon area 3 (mm ²)	Tendon area 4 (mm ²)	Tendon area 5 (mm ²)	Time (years)	Static disp. (mm)	Dynamic disp. (mm)
Distribution	Gumbel	Gumbel	Gumbel			
CoV	0.1	0.1	0.1			
Mean	2250α	2250α	2250α		17.6	23.9
Min	1444α	1474α	1472α	0	15.6	19.7
Max	3388α	3455α	3370α	80	23.5	27.7

α is the time-varying ratio between corroded area over initial area of tendon following Fig. 12.

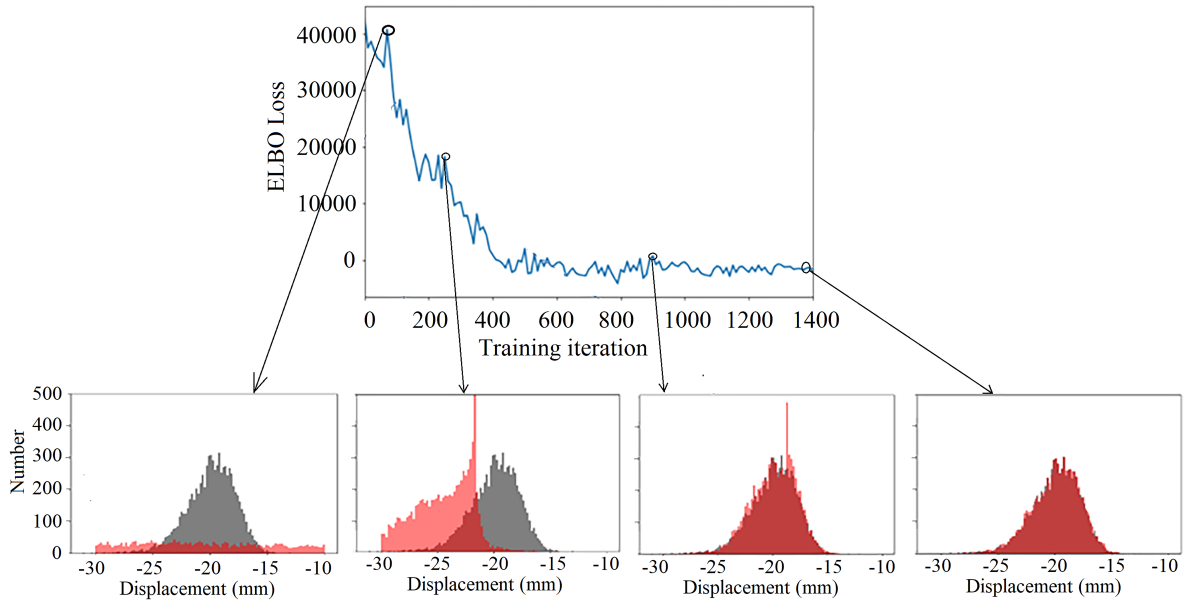


Figure 13: Evolution of training loss with respect to the number of iteration for BNN models predicting maximum static deflection of bridge. At the bottom, the predicted values of BNN models are compared with those from FEM models.

computers. There are in total $9E4$ simulations with random variable inputs sampled from distributions specified in Table 5. For each simulation, two types of analysis were performed: the static analysis in which the truck stays immobile at the middle of the bridge and the dynamic analysis where the truck crosses along the bridge. The output of the analyses is the vertical displacement at the middle of the bridge. Afterwards, two variants of BNN-base surrogate models are designed, corresponding to these two types of analysis.

After accomplishing the suite of simulations, input parameters and corresponding outputs form a tabular database of $9E4$ lines and 10 fields of values, which is split into training, validation, and testing dataset with the ratio of 80/10/10,

serving for the development of BNN mode. Figure 13 shows the evolution of ELBO loss versus the number of training iteration, a.k.a epochs when predicting the maximum static deflection under a concentrated force equivalent to a HS20 truck placed at the middle of the bridge. Notice that the loss sharply decreases for the first 400 epochs, then fluctuates considerably before becoming more stable from around epoch 1200. To show the significance of the loss function more clearly, the corresponding histogram of deflection values predicted by BNN (in the gray region) for the training dataset is plotted and overlapped on those obtained from FEM models (in red region). By doing so, one can monitor the whole 9×10^4 results and understand better how the model

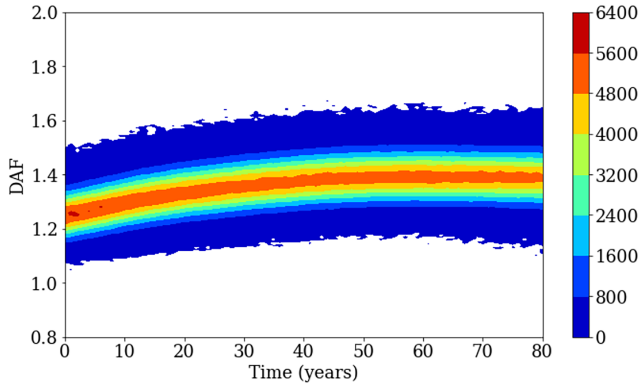


Figure 14: Contour plot of the evolution of DAF's histogram versus working years.

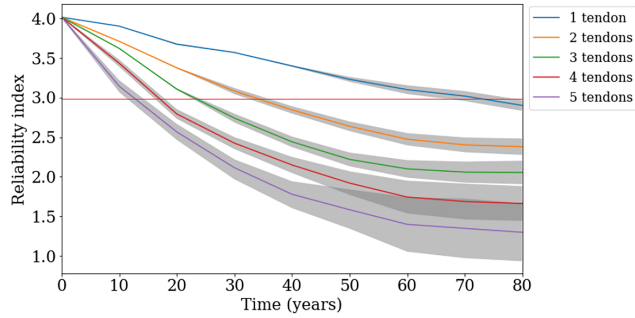


Figure 15: Time-varying of reliability index for various corrosion scenarios

improves during the training process. Initially, the model provides results randomly distributed as shown by the nearly flat-shaped red region in the left-most subfigure. Progressively, the shape of the distribution of outputs approaches that of the FEM results, with the highest peak near the mean value of the FEM distribution (17.5 mm). Finally, a good consistency is achieved between the two results in both shape and amplitude, as depicted in the right-most subfigure. By analogy, a second BNN model is constructed to estimate the bridge's maximum deflection when a HS20 truck crosses the bridge. Further detail of how to simulate numerically a bridge under moving load can be found in [9].

Having trained the two BNN models, one for maximum static deflection and the other for maximum dynamic deflection, they are employed to predict the reliability of the bridge with respect to the corrosion of tendons over time. Different corrosion scenarios are taken into account where the number of damaged tendons varies from zero to five. For each scenario, a 9×10^6 -sample testing data corresponding to a service life between [0,10,..., 80] years, i.e., 10^6 samples for each year, is generated with random parameters after Table 5. Fig. 14 illustrates a contour plot of the evolution of the histogram of DAF defined by Eq. (28) for a 5-tendon corrosion scenario in which the number of samples is color-coded and annotated by the color bar on the right. Then, a threshold value of DAF is defined by Eurocode [11] to derive the failure probability and reliability index of the structure as

below:

$$g_{DAF} = DAF_{limit} = 1.5 \quad (29)$$

Computed reliability results are graphically illustrated in Fig. 15 in which colored lines denote mean values of the reliability index, and shaded areas indicate the confidence area of prediction. For example, the purple line is directly derived from Fig. 14 using $DAF_{limit} = 1.5$, whereas the shaped area is obtained by repeating 100 times analogous calculations with other sets of BNN' weights drawn from the posterior distributions. In the initial year, there is supposedly no corrosion occurring. Thus, the reliability indexes for all cases are logically the same, i.e., $\beta = 4.0$. After that, a descending trend of β is apparently observed as corrosion increases over time. Moreover, the more tendons are corroded, the higher failure probability is, leading to the lower reliability index, for example, β decreases from 4.0 to 2.9, 2.4, 2.1, 1.6, and 1.3 after 80 years for 1, 2, 3, 4 and 5-corroded tendon cases, respectively. It is also noted that the uncertainty rises with the extent of corrosion as the shaded area widens significantly. These results are useful for determining the remaining useful life and setting an optimal maintenance plan. For example, to maintain a highly reliable service, i.e., ensure that the failure probability is only around 1/1000 ($\beta \sim 3.0$) in terms of DAF, one draws a horizontal line at 3.0-level in Fig. 15. The figure suggests that the structure needs to be checked after 11 years in operation for 5-tendon corrosion cases (already accounting for the confidence interval). In contrast, if in a favorable environment and well-protected tendon, the reliability can be ensured up to 70 years (1 tendon corrosion case).

In terms of time complexity, the FEM takes around 10 hours of computation on a server of 12 cores, 64 Gb and GPU 2080Ti, to accomplish the $9E4$ -sample database. After training with this database, the BNN-based method only requires 45 minutes for a testing database of $9E6$ samples, meaning a reduction of three orders of magnitude in time complexity is obtained. Moreover, the BNN model can be deployed in any portable laptop rather than a high-performance server. These results emphasize the application of the BNN-based method in complementing the time-varying structural reliability analysis of complex structures where the conventional FEM is prohibited because of a large required number of calculations and high computational cost. On the other hand, confidence intervals provided by the BNN help engineers to have some ideas about the bound, i.e., confidence limits of prediction results, thus avoid excessively under or over-conservative assessments about the actual state of the structure.

4. Conclusion

This study proposes a BNN-based method to estimate structures' time-varying reliability and associated uncertainty estimation with significantly reduced time complexity. It is noted that performing dynamic reliability analysis and estimating the uncertainty of analysis results are two tedious

tasks; each requires repeating analyses multiple times. Here, the proposed method can accomplish these two tasks at the same time, yielding information about both aleatory and epistemic uncertainty, which is yet accomplished by neither traditional DL-based surrogate models nor the conventional Monte Carlo simulation based on FEM. It can be used as a promising alternative, providing thorough and broad understandings of structures' static and dynamic responses.

These mentioned statements are supported through three case studies ranging from a theoretical problem to a 2D steel frame structure and a 3D model of a real-world bridge structure. Though the time complexity of the BNN model is higher than that of analytical solutions, it is significantly lower than model-based simulation, up to a reduction of three orders of magnitude for 3D simulation. Thanks to these superior properties, various applications of the BNN model can be realized, such as: providing informative results even with a low number of available data as in case study 1, performing analysis with different load conditions (case study 2), and different corrosion scenarios (case study 3).

Although the proposed method can be applied for different structures, the adequate neural network architecture for each problem, including the number of layers and perceptron, needs to be redetermined through a preliminary parametric study. Thus, it is desired to seek an auto-machine learning algorithm capable of automatically adapting to the problem of interest. Another aspect of improvement is to consider multi-limit states simultaneously as done in the standard design practice. This can be done by using a hybrid learning loss function combining different terms, each representing a limit state.

In the future, it is noteworthy to extend the BNN to perform reliability analysis of structure considering simultaneously multi-limit states, which are not uncommon in real scenarios. Another interesting direction is to build a BNN model accounting for stochastic excitation such as wind loading, ground motion, and non-linear behavior of structures, where the lack of data is still problematic for other deep learning-based methods.

Acknowledgement

This work is supported in part by a UK India Education and Research Initiative (UKIERI) Grant 'DST UKIERI-2018-19-011' and in part by an Institutional Links grant, ID 429715093, under the Newton Programme Vietnam partnership. The grant is funded by the UK Department of Business, Energy and Industrial Strategy (BEIS) and delivered by the British Council.

References

- [1] AASHTO, 1998. AASHTO LRFD Bridge Design Specifications. American Association of State Highway and Transportation Officials.
- [2] Alvarez-Ruso, L., Graczyk, K.M., Saul-Sala, E., 2019. Nucleon axial form factor from a bayesian neural-network analysis of neutrino-scattering data. *Phys. Rev. C* 99, 025204.
- [3] Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13, 281–305.
- [4] Bingham, E., Chen, J.P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., Goodman, N.D., 2019. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.* 20, 973–978.
- [5] Bishop, C.M., 1997. Bayesian neural networks. *J. Braz. Comput. Soc.* 4.
- [6] Burkov, A., 2019. The hundred-page machine learning book. Andriy Burkov Quebec City, Can.
- [7] Cheng, J., Li, Q., Xiao, R.C., 2008. A new artificial neural network-based response surface method for structural reliability analysis. *Probabilistic Engineering Mechanics* 23, 51–63.
- [8] Committee, A., 2019. Building code requirements for structural concrete (aci 318-08) and commentary, American Concrete Institute.
- [9] Dang, V.H., Ha, M.H., 2020. Direct numerical simulation of dynamic response of bridges subjected to vehicular loads. *Int. J. Dyn. Control.* , 1–12.
- [10] Du, W., Luo, Y., Wang, Y., 2019. Time-variant reliability analysis using the parallel subset simulation. *Reliability Engineering & System Safety* 182, 250–257.
- [11] Eurocode, C., 1991. Actions on structures, part 2: Traffic loads on bridges. Brussels: European Standard EN 2, 2003.
- [12] Fan, X., Liu, Y., 2019. Use of monitored daily extreme stress data for performance prediction of steel bridges: Dynamic linear models and gaussian mixed particle filter. *Mech. Syst. Signal Process.* 121, 841–855.
- [13] Fang, T., Jiang, C., Huang, Z., Wei, X., Han, X., 2018. Time-variant reliability-based design optimization using an equivalent most probable point. *IEEE Trans. Rel* 68, 175–186.
- [14] Han, X., Yang, D.Y., Frangopol, D.M., 2019. Time-variant reliability analysis of steel plates in marine environments considering pit nucleation and propagation. *Probabilistic Eng. Mech.* 57, 32–42.
- [15] Huang, Z., Tu, Y.M., Grip, N., Sabourova, N., Bagge, N., Blanksvärd, T., Ohlsson, U., Elfgrén, L., 2016. Modelling of damage and its use in assessment of a prestressed concrete bridge, in: 19th IABSE Congress, Stockholm 2016, September 21–23, pp. 2093–2108.
- [16] Huh, J., Haldar, A., 2011. A novel risk assessment for complex structural systems. *IEEE Trans. Rel* 60, 210–218.
- [17] Jiang, C., Huang, X., Han, X., Zhang, D., 2014. A time-variant reliability analysis method based on stochastic process discretization. *J. Mech. Des.* 136.
- [18] Kalatjari, V., Kaveh, A., Mansoorian, P., 2011. System reliability assessment of redundant trusses using improved algebraic force method and artificial intelligence.
- [19] Kaveh, A., Dadras Eslamlou, A., 2019. An efficient method for reliability estimation using the combination of asymptotic sampling and weighted simulation. *Scientia Iranica* 26, 2108–2122.
- [20] Kaveh, A., Eslamlou, A.D., Javadi, S., Malek, N.G., 2021. Machine learning regression approaches for predicting the ultimate buckling load of variable-stiffness composite cylinders. *Acta Mechanica* 232, 921–931.
- [21] Kaveh, A., Ilchi Ghazaan, M., 2015. Structural reliability assessment utilizing four metaheuristic algorithms. *Iran University of Science & Technology* 5, 205–225.
- [22] Kaveh, A., Javadi, S.M., Moghanni, R.M., 2020. Reliability analysis via an optimal covariance matrix adaptation evolution strategy: Emphasis on applications in civil engineering. *Periodica Polytechnica Civil Engineering* 64, 579–588.
- [23] Kingma, D.P., Salimans, T., Welling, M., 2015. Variational dropout and the local reparameterization trick, in: *Proc. NeurIPS*, pp. 2575–2583.
- [24] Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M., 2017. Automatic differentiation variational inference. *J. Mach. Learn. Res.* 18, 430–474.
- [25] Kullback, S., 1997. Information theory and statistics. Courier Corporation.
- [26] Liu, M., Chowdhary, G., Castrada da Silva, B., Liu, S., How, J.P., 2018. Gaussian processes for learning and control: A tutorial with examples. *IEEE Control Systems Magazine* 38, 53–86.

- [27] Lu, Z.H., Cai, C.H., Zhao, Y.G., Leng, Y., Dong, Y., 2020. Normalization of correlated random variables in structural reliability analysis using fourth-moment transformation. *Struct. Saf.* 82, 101888.
- [28] Lu, Z.H., Leng, Y., Dong, Y., Cai, C.H., Zhao, Y.G., 2019. Fast integration algorithms for time-dependent structural reliability analysis considering correlated random variables. *Struct. Saf.* 78, 23–32.
- [29] McKenna, F., 2011. Opensees: a framework for earthquake engineering simulation. *Comput. Sci. Eng.* 13, 58–66.
- [30] Paisley, J., Blei, D., Jordan, M., 2012. Variational bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430*.
- [31] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library, in: *Proc. NeurIPS*, pp. 8024–8035.
- [32] Paultre, P., Chaallal, O., Proulx, J., 1992. Bridge dynamics and dynamic amplification factors—a review of analytical and experimental findings. *Can. J. Civ. Eng.* 19, 260–278.
- [33] Podolny Jr, W., 1992. Corrosion of prestressing steels and its mitigation. *PCI Journal* 37, 34–55.
- [34] Simulia, D., 2016. Abaqus theory guide. *dassault systèmes*.
- [35] Straub, D., Schneider, R., Bismut, E., Kim, H.J., 2020a. Reliability analysis of deteriorating structural systems. *Struct. Saf.* 82, 101877.
- [36] Straub, D., Schneider, R., Bismut, E., Kim, H.J., 2020b. Reliability analysis of deteriorating structural systems. *Struct. Saf.* 82, 101877.
- [37] Su, G., Peng, L., Hu, L., 2017. A gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. *Struct. Saf.* 68, 97–109.
- [38] Sun, W., Zamani, M., Hesamzadeh, M.R., Zhang, H., 2020. Data-driven probabilistic optimal power flow with nonparametric bayesian modeling and inference. *IEEE Trans. Smart Grid* 11, 1077–1090.
- [39] Val, D.V., Melchers, R.E., 1997. Reliability of deteriorating rc slab bridges. *J. Struct. Eng.* 123, 1638–1644.
- [40] Wang, Z., Chen, W., 2017. Confidence-based adaptive extreme response surface for time-variant reliability analysis under random excitation. *Struct. Saf.* 64, 76–86.
- [41] Xiang, Z., Chen, J., Bao, Y., Li, H., 2020. An active learning method combining deep neural network and weighted sampling for structural reliability analysis. *Mech. Syst. Signal Process.* 140, 106684.
- [42] Xiao, S., Lu, Z., 2017. Structural reliability sensitivity analysis based on classification of model output. *Aerosp. Sci. Technol.* 71, 52–61.
- [43] Yin, T., Zhu, H.P., 2020. An efficient algorithm for architecture design of bayesian neural network in structural model updating. *Comput-Aided Civ. Inf.* 35, 354–372.
- [44] Zhao, E., Wu, C., 2021. Long-term safety assessment of large-scale arch dam based on non-probabilistic reliability analysis, in: *Structures*, Elsevier. pp. 298–312.