

## **ROBUST ARTIFICIAL NEURAL NETWORK FOR RELIABILITY ANALYSIS**

**Uchenna Oparaji<sup>1,2</sup>, Rong-Jiun Sheu<sup>2</sup>, and Edoardo Patelli<sup>1</sup>**

<sup>1</sup>Institute for Risk and Uncertainty, University of Liverpool  
Chadwick Building, Peach Street, Liverpool L69 7ZF, United Kingdom  
e-mail: {u.oparaji, epatelli}@liverpool.ac.uk

<sup>2</sup> Institute of Nuclear Engineering and Science  
National Tsing Hua University, Hsinchu, Taiwan  
e-mail: {rjsheu}@mx.nthu.edu.tw

**Keywords:** Artificial Neural Network, Uncertainty Quantification, Reliability Analysis.

**Abstract.** *Artificial Neural Networks (ANN) are used in place of expensive models to reduce the computational burden required for reliability analysis. Often, ANNs with selected architecture are trained with the back-propagation algorithm from few data representatives of the input/output relationship of the underlying model of interest. However, different performing ANNs might be obtained from the same training data, leading to an uncertainty in selecting the best performing ANN. On the other hand, using cross-validation to select the best performing ANN based on the highest  $R^2$  value can lead to a biasing in terms of the prediction made by the selected ANN. This is due to the fact that the use of  $R^2$  cannot determine if the prediction made by ANN is biased. Additionally,  $R^2$  does not indicate if a model is adequate, as it is possible to have a low  $R^2$  for a good model and a high  $R^2$  for a bad model. Hence we propose an approach to improve the prediction robustness of an ANN based on coupling Bayesian framework and model averaging technique into a unified framework. The model uncertainties propagated to the robust prediction is quantified in terms of confidence intervals. Two examples are used to demonstrate the applicability of the approach*

## 1 INTRODUCTION

Nowadays, numerical models are increasingly used to analyze and predict the performance of complex critical systems. Concurrently, engineering practitioners are concerned with uncertainty, which is inherent to these systems. As a consequence, probabilistic analyses, such as reliability analysis [1], robust design optimization [2], and sensitivity analysis [3], have received much attention in the last decades. However, the computational cost required for performing the aforementioned analyses depends on several factors such as: the numerical model representing the system, the type of analysis, and the treatment of uncertainties (i.e. aleatory and/or epistemic uncertainty). In the context of reliability analysis, the propagation of parameter uncertainties from model inputs to outputs is performed by means of Monte Carlo simulation based approaches. These simulation approaches include: Monte Carlo (MC) [4], and advanced MC such as: Importance Sampling [5], Directional Sampling [6], Line Sampling [7], Subset Simulation [8] etc. Although, advanced MC methods are very efficient, the computational cost required to perform reliability analysis is usually expensive. A popular strategy to reduce the computational costs is to replace the real model with a surrogate model such as an artificial neural network (ANN). ANNs can be constructed based on few data sets from the underlying model of interest. On the other hand, the use of an ANN for this kind of analysis introduces model selection uncertainty in addition to biasing and variance in the estimated quantity of interest. As a matter of fact, an ANN with a specific architecture trained repeatedly with a finite data set  $D_{train}(\mathbf{x}, \mathbf{y})$  results to different performing ANNs whose cost functions are being trapped at different local minima of the cost function solution space. This phenomenon occurs as a result of the random initialization of the weights within each ANN. Consequently, it is of common practice to select the best ANN from the uncertain set on the basis of performance on an independent validation set, and to keep only the network with the lowest validation error and discard the rest. However, there are two disadvantages to such approach. Firstly, all of the effort required to train the remaining networks is wasted. Secondly, the generalization performance of the networks on the validation set has a random component due to the noise on the data, hence the network which had the lowest error on the validation set might perform poorly on a new test set. These disadvantages can be overcome by combining the networks together to form a committee that can significantly improve the robustness of the predicted quantity. Hence, in this paper an approach is proposed to improve the robustness of a neural network when used to predict the probability of failure  $p_F$ . The outline of this paper is as follows: In Section 2, a succinct theory of reliability analysis using simulation approach and neural network modelling is discussed. This is followed by the proposed approach (Section 3). Next, to demonstrate the applicability of the proposed approach, two numerical examples are tested in Section 4. Finally, conclusions are provided in Section 5.

## 2 RELIABILITY ANALYSIS

The limit-state function can simply be defined as a deterministic mapping from the  $z$ -dimensional input space to a one-dimensional output space:

$$G : \mathbf{x} \in D_{\mathbf{x}} \subset \mathbb{R}^z \rightarrow y = G(\mathbf{x}) \in \mathbb{R} \quad (1)$$

where  $\mathbf{x}$  is the  $z$ -dimensional state variables and  $y$  the performance variable.  $G(\mathbf{x})$  indicates if a realization  $\mathbf{x} \in D_{\mathbf{x}}$  corresponds to the safe state ( $G(\mathbf{x}) > 0$ ) or failed state ( $G(\mathbf{x}) \leq 0$ ). In the context of probability theory, the failure probability,  $p_F$ , is defined as the probability that a

realization  $\mathbf{x} \in D_{\mathbf{x}}$  corresponds to a failed state in terms of the limit-state function  $G(\mathbf{x})$ :

$$p_F = \mathbb{P}(G(\mathbf{x}) \leq 0) = \int_{D_f} f_X(\mathbf{x}) d\mathbf{x} \quad (2)$$

where  $D_f = \mathbf{x} \in D_{\mathbf{x}} : G(\mathbf{x}) \leq 0$  is the failure region and  $f_X(\mathbf{x})$  is the joint probability density function of the state variables  $X$ . As Eq.(2) is analytically intractable due its multidimensional nature, Monte Carlo simulation (MCS) (see [4]) allows on to numerically compute the estimate of the failure probability  $p_F$ , considering a large sample of size  $N$  :

$$\hat{p}_F = \frac{1}{N} \sum_{i=1}^N \mathbb{I}_{G(\mathbf{x}) \leq 0}(x_i) \quad (3)$$

where  $\mathbb{I}_{G(\mathbf{x}) \leq 0}$  is the indicator function for failure such that  $\mathbb{I} = 1$  for  $G(\mathbf{x}) \leq 0$  and  $\mathbb{I} = 0$  otherwise.

## 2.1 MODELLING ARTIFICIAL NEURAL NETWORK FOR RELIABILITY ANALYSIS

A setback on the use of MCS to compute the estimate of  $p_F$  is the large number of model evaluation required for computing a robust estimate. Hence, an ANN can be used in place of the limit state function to reduce the computational cost. The construction an ANN requires a set of real-valued input/output data pairs  $D_{train}(\mathbf{x}, \mathbf{y})$  of size  $N_{train}$  generated according to a signal plus noise model  $\mathbf{y} = \boldsymbol{\mu}(\mathbf{x}) + \boldsymbol{\varepsilon}$ , where  $\mathbf{y}$  is the observed performance generated from the expensive model,  $\mathbf{x}$  is the independent state variables sampled from a joint probability density  $\Omega(\mathbf{x})$ ,  $\boldsymbol{\varepsilon}$  is independent, identically distributed (iid) noise sampled from a density  $\Psi(\boldsymbol{\varepsilon})$  (not necessarily Gaussian) having mean of 0 and variance  $\sigma^2$ , and  $\boldsymbol{\mu}(\mathbf{x})$  the unknown function that is needed to be approximated by finding an approximation  $\hat{\boldsymbol{\mu}}(\mathbf{x})$  from  $D_{train}(\mathbf{x}, \mathbf{y})$ . A priori assumptions can be made about the functional form of  $\boldsymbol{\mu}(\mathbf{x})$ . However, since a parametric function class is usually unknown, non-parametric regression approach must be resorted to. Using the non-parametric approach, one constructs an estimate  $\hat{\boldsymbol{\mu}}(\mathbf{x})$  of  $\boldsymbol{\mu}(\mathbf{x})$  from a large class of functions  $\Upsilon$  known to have good approximation properties. The class of approximation functions usually contains a set of estimators  $f(w, \mathbf{x}) \subset \Upsilon$  for which the elements of each subclass  $f(w, \mathbf{x})$  are continuously parametrized by a set of  $p$  weights  $w^\alpha$ ;  $\alpha = 1, 2, \dots, p$ . The gradient decent algorithm [9] which is used to minimize the cost function  $J$  of the neural network defined as:

$$J = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} (y_i - \hat{y}_i)^2 \quad (4)$$

by finding a set of weights  $w^\alpha$  such that for any given input, the cost function is sufficiently small. However, a limitation of the gradient decent algorithm to train an ANN is the possibility of the cost function to be trapped in a local minimum, thereby reducing the predictive capability of the network.

## 3 THE PROPOSED APPROACH

The proposed approach aims towards improving the robustness of the prediction made by an ANN when used to perform reliability analysis [10]. The underlying idea behind the proposed approach is to construct a set of ANNs with the same architecture and based on the same training data set  $D_{train}(\mathbf{x}, \mathbf{y})$ . By doing so, a distribution of identical ANNs having their error functions

trapped in different local minima is created. The major highlight of this approach is that the solution space of the error function is exploited as many times as possible with the possibility of locating a global minima on the error surface. Further, Bayes' theorem is used to evaluate the posterior probability of each of the trained ANN based on their likelihood to predict the training data. This is followed by the use of a model averaging technique (adjustment factor approach see [11]) to combine the total prediction made by all the ANNs in the set to yield a robust prediction that converges to the true value. Finally, the model uncertainty propagated to the predicted quantity is quantified in terms of confidence intervals.

### 3.1 BAYESIAN MODEL SELECTION FOR ARTIFICIAL NEURAL NETWORK

Given a set of  $M$  identical (i.e. the same model structure) competing ANNs  $N_k, k = 1, 2, \dots, M$  trained with same data set  $D_{train}(\mathbf{x}, \mathbf{y})$ , Bayes theorem can be used to express the posterior probability of the  $k^{th}$  ANN in the set which is defined by:

$$P(N_k|D_{train}) = \frac{P(D_{train}(\mathbf{x}, \mathbf{y})|N_k)P(N_k)}{\sum_{q=1}^M P(D_{train}(\mathbf{x}, \mathbf{y})|N_q)P(N_q)} \quad (5)$$

where  $P(D_{train}(\mathbf{x}, \mathbf{y})|N_k)$  is the likelihood of training data  $D_{train}(\mathbf{x}, \mathbf{y})$  for the  $N_k$  ANN, and  $P(N_k)$  is the prior probability of  $N_k$ , which is the ANN probability evaluated before observing training data  $D_{train}(\mathbf{x}, \mathbf{y})$ . The prior ANN probability  $P(N_k)$  can be specified depending on the existing prior knowledge about the credibility of ANN  $N_k$ , or it can be given as a uniform probability,  $P(N_k) = 1/M$ , if no additional information is provided. The advantage of assigning uniform prior probability to  $P(N_k)$  is that the difficulty of estimating the prior probability numerically is avoided. The likelihood  $P(D_{train}(\mathbf{x}, \mathbf{y})|N_k)$  may be thought of as the probability of observing the training data  $D_{train}(\mathbf{x}, \mathbf{y})$  under  $N_k$  ANN. It supplies a relative measure of how well the  $N_k$  ANN is supported by the training data  $D_{train}(\mathbf{x}, \mathbf{y})$ . Since the denominator in Eq.(5) is common for all the ANNs, the posterior ANN probability is proportional to prior probability and the likelihood. The likelihood of each ANN is evaluated by measuring the degree of agreement between the training data  $D_{train}(\mathbf{y})$  and the response  $\hat{y}$  for each ANN. Hence, a probabilistic relationship between training data  $D_{train}(\mathbf{x}, \mathbf{y})$  and ANN predictions  $\hat{y}$  involving uncertainty can be described. Typically, the bias function and noise are included as parts of the probabilistic relationship to match ANN predictions with training data. The bias function captures the discrepancies between the expensive model responses and predictions made by the ANN. The noise is usually assumed to be independent and identically distributed normal random variable with a mean of zero [12]. Various authors, see e.g. [13, 14, 15] have used the Bayesian statistical methodology to quantify the uncertainty in the bias function modelled as a Gaussian process. In their works, a mathematical formulation that combines bias function associated with the ANN and noise from training data is utilized to describe the probabilistic relationship between the training data  $D_{train}(\mathbf{x}, \mathbf{y})$  and ANN predictions  $\hat{y}$ . The mathematical formulation of this probabilistic relationship is given by the following equation:

$$D_{train}(\mathbf{y}) = \hat{y} - \varepsilon \quad (6)$$

where  $\varepsilon$  is a random variable that covers both bias associated with the ANN prediction  $\hat{y}$  and the noise in the response training data  $D_{train}(\mathbf{y})$ .  $\varepsilon$  is assumed to be an independent identically distributed random variable with a mean  $\mu$  of zero. The use of  $\varepsilon$  with zero mean does not shift ANN prediction  $\hat{y}$ . This reflects the fact that  $\hat{y}$  is the most probable prediction value for the ANN. The bias function is not included as a separate term in the probabilistic relationship. This

is due to the fact that introducing a separate bias function results in shifting the prediction  $\hat{y}$  of the ANN from the initially predicted value.

The likelihood  $P(D_{train}(\mathbf{x}, \mathbf{y})|N_k)$  of training data  $D_{train}(\mathbf{x}, \mathbf{y})$  for ANN  $N_k$  is evaluated by observing where the training data points  $D_{train}(\mathbf{y})$  are located in the distribution of  $\hat{y}$  estimated by  $N_k$ . The procedures to estimate the distribution  $P(\hat{y}|N_k)$  of  $N_k$  and the likelihood  $P(D_{train}(\mathbf{x}, \mathbf{y})|N_k)$  is given. First, the uncertainty in errors of predictions  $\hat{y}$  made by  $N_k$  is quantified by introducing an assumption that the prediction errors are independent and identically distributed normal random variable with a mean  $\mu$  of zero. The error of the prediction of the  $k^{th}$  network is represented by the following:

$$\epsilon_{ki} = D_{train}(y_i) - \hat{y}_i, \epsilon_{ki} \sim N(0, \sigma_k^2), i = 1, 2, \dots, N \quad (7)$$

where  $D_{train}(y_i)$  is the  $i^{th}$  training response output data,  $\hat{y}_i$  the prediction of the training data made by  $N_k$ ,  $\sigma_k^2$  is the variance of prediction error  $\epsilon_{ki}$ , and  $N$  the number of samples in the training data. The prediction error  $\epsilon_{ki}$  measured is considered to be a random sample from a normal distribution with a mean  $\mu$  of zero and variance  $\sigma_k^2$ . Using the principle of maximum likelihood estimation (MLE) (see [16]), the variance  $\sigma_k^2$  for  $N_k$  can be estimated as:

$$\sigma_k^2 = \frac{1}{N} \sum_{i=1}^N \epsilon_{ki}^2 \quad (8)$$

Secondly, the predictive distribution  $P(\hat{y}|N_k)$  of response  $\hat{y}$  under model  $N_k$  is created by including the prediction error obtained in the previous step into the prediction of  $\hat{y}$  made by  $N_k$ . This predictive distribution is defined by the following equation:

$$P(\hat{y}|N_k) = D_{train}(\mathbf{y}) + \epsilon_{ki} \quad (9)$$

Lastly, assuming that the residuals between the training data  $D_{train}(\mathbf{x}, \mathbf{y})$  and ANN  $N_k$  output  $\hat{y}$  are normally and independently distributed with a mean of zero and constant variance  $\sigma_k^2$ , the likelihood function  $P(D_{train}(\mathbf{x}, \mathbf{y})|N_k)$  is approximated by:

$$P(D_{train}(\mathbf{x}, \mathbf{y})|N_k) \approx \frac{1}{\sqrt{2\pi\sigma_k^2}} \frac{1}{N} \sum_{i=1}^N \exp\left\{-\frac{[y_i - \hat{y}_{ki}]^2}{2\sigma_k^2}\right\} \quad (10)$$

### 3.2 ROBUST ARTIFICIAL NEURAL NETWORK PREDICTION

To obtain a robust prediction from an ANN, the estimates made by all the subsequent trained ANNs are combined using model averaging technique. Specifically, the adjustment factor approach (see [11]) which is a model averaging technique is combined with Bayes' theorem. In this way, the ANN having the highest posterior probability is used in conjunction with other respective ANNs trained to correct the bias estimate predicted by the single ANN. The adjustment factor  $A_f$  is evaluated by assuming the error between the prediction of all the subsequent trained ANNs and the training data are normally distributed. The robust ANN prediction can be obtained from the following equation:

$$y_{robust} = \hat{y}^* + A_f \quad (11)$$

where  $\hat{y}^*$  represents the point estimate of the best ANN in the set characterised by the highest probability, and  $y_{robust}$  represent the robust prediction which incorporates the model uncertainty.

Since the adjustment factor  $A_f$  is assumed to be a normal distribution, the expected value and variance of the adjustment factor  $A_f$  is given by the following relationships:

$$E(A_f) = \sum_{k=1}^M P(N_k | D_{train}) (\hat{y}_k - \hat{y}^*) \quad (12)$$

$$Var(A_f) = \sum_{k=1}^M P(N_k | D_{train}) (\hat{y}_k - E(y_{robust}))^2 \quad (13)$$

Similarly, the expected value and variance of the robust prediction  $y_{adj}$  can be estimated from the following relationships:

$$E(y_{robust}) = \hat{y}^* + E(A_f) \quad (14)$$

$$Var(y_{robust}) = Var(A_f) \quad (15)$$

where  $E(A_f)$  and  $Var(A_f)$  represents the expected value and variance of the adjustment factor, and  $E(y_{robust})$  and  $Var(y_{robust})$  represents the expected value and variance of the robust estimate.

### 3.3 CONFIDENCE INTERVAL FOR ROBUST ESTIMATE

To quantify the uncertainty in the robust prediction  $y_{robust}$  due to model uncertainty, confidence intervals are established. In particular, the 5<sup>th</sup> and 95<sup>th</sup> percentiles of the robust prediction are used to quantify the model uncertainty. In theory, this interval is likely to contain the true estimated value. As the model uncertainty is assumed to follow normal distribution, the confidence intervals (see [17]) are calculated from the following equations:

$$\overline{CI} = E(y_{robust}) + z^* \sqrt{Var(y_{robust})} \quad (16)$$

$$\underline{CI} = E(y_{robust}) - z^* \sqrt{Var(y_{robust})} \quad (17)$$

where  $\overline{CI}$  and  $\underline{CI}$  represents the upper and lower confidence intervals of the robust estimate and  $z^*$  represents the upper critical value of the Gaussian distribution quantifying the model uncertainty.

## 4 NUMERICAL EXAMPLE

### 4.1 THE HAT FUNCTION

The hat function is defined by the analytical expression [18]:

$$G(\mathbf{x}) = 20 - (x_1 - x_2)^2 - 8(x_1 + x_2 - 4)^3 \quad (18)$$

where  $x_i, i = 1, 2$  is defined as Gaussian variables with mean  $\mu_{x_i} = 0.5$  and standard deviation  $\sigma_{x_i} = 1.0$ . Failure is defined as  $G(\mathbf{x}) \leq 0$  hence  $p_F = \mathbb{P}(G(\mathbf{x})) \leq 0$ . The limit state surface plot of the hat function is shown in Fig. 1.

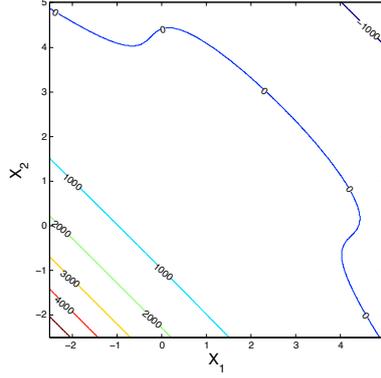


Figure 1: Limit State Surface of Hat Function

The aim of this example is to verify the proposed approach by replacing the limit state function with an ANN, then compute a robust estimate of  $\hat{p}_F$ , quantify the model uncertainty and, finally, verify the number of identical ANNs that must be trained to attain an optimal robust estimate of  $\hat{p}_F$ .

## 4.2 ANALYSIS 1

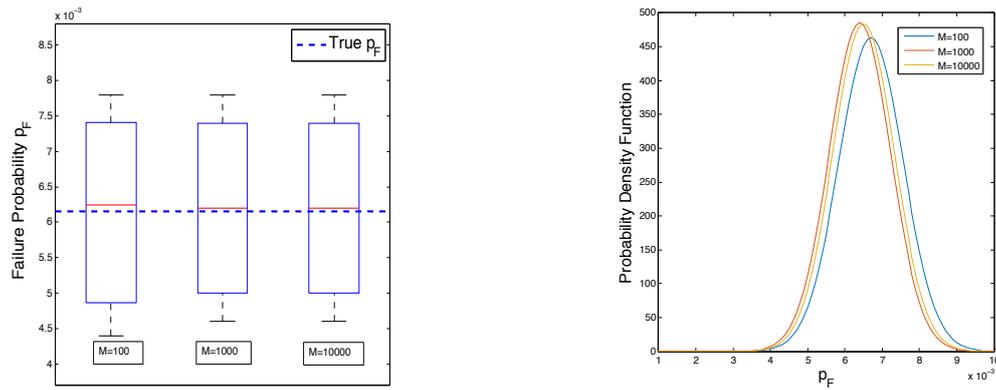
Training samples  $D_{train}(\mathbf{x}, \mathbf{y})$  of size  $N_{train} = 2000$  have been generated via Latin hypercube sampling (LHS) algorithm[19] from the hat function. Two sets  $Z_1 = N_k, k = 1, 2, \dots, M$  and  $Z_2 = N_i, i = 1, 2, \dots, M$  composed of  $M = 1000$  identical ANNs have been trained based on  $D_{train}(x, y)$ . Specifically, in the first set ( $Z_1$ ), all the training samples in  $D_{train}(x, y)$  have been used to train the ANNs to maximize their predictive performances. For the second set  $Z_2$ , 80% of the training samples  $D_{train}$  have been used to train the ANNs and the remaining 20% used for validation. The network architecture chosen for the ANNs in both sets composed of three hidden layers (2,7,1). Next, the posterior probability of the ANNs in set  $Z_1$  has been estimated using Bayes' formula by assigning uniform prior probability  $P(N_k) = 1/M$  to each ANN. On the other hand, the coefficient of determination  $R^2$  for the ANNs in set  $Z_2$  have been estimated based on the 20% validation samples. Table.1 shows a comparison of 10 selected ANNs from  $Z_1$  and  $Z_2$  based on their posterior probabilities and their error values  $R^2$ . It should be noted that  $i^{th}$  ANN in both set ( $Z_1$  and  $Z_2$ ) have been trained inside the same iteration loop, hence the initialization of the weight values within each loop it is assumed to be similar. Therefore, their resultant performances are expected to be similar. As shown in Table.1, although the ANNs  $N_i, i = 1, 2, \dots, M$  in sets  $Z_1$  and  $Z_2$  are identical as they have been trained in the same iteration loop, the performance measures in terms of the posterior probability and  $R^2$  shows no agreement. For example, the 6<sup>th</sup> and 10<sup>th</sup> ANNs have the highest posterior probability, however their corresponding  $R^2$  values don't show a similar trend. Hence, we can support our claim that the use of  $R^2$  value to select the best model is a biased method. Further, to implement the proposed approach, the ANNs in  $Z_1$  have been chosen as they have better performance (i.e. due more samples used to train them). To accurately compute a robust estimate of  $\hat{p}_F$ ,  $10^4$  Monte Carlo simulation runs have been used for each ANN, and the proposed approach presented have been used to average out the prediction made by each ANN model into a robust value that is converges to the true value. Finally, the model uncertainty propagated to robust prediction of  $\hat{p}_F$  has been quantified in terms of confidence intervals estimated.

ANN	$P(D_{train} N_k)$	$R^2$
1	0.103	0.9989
2	0.087	0.9998
3	0.101	0.9993
4	0.117	0.9999
5	0.070	0.9998
6	0.122	0.9998
7	0.093	0.9996
8	0.099	0.9999
9	0.086	0.9999
10	0.122	0.9995

Table 1: Artificial Neural Networks Posterior Probability Calculated Compared to Corresponding  $R^2$  value.

### 4.3 ANALYSIS 2

To check the number of ANNs that must be trained in order to obtain a robust value (i.e. close to reference value), the real model has been used to estimate the reference value of  $\hat{p}_F$  adopting the same failure criteria (i.e  $G(\mathbf{x}) \leq 0$ ) and  $N = 10^4$  samples. On the other hand, 3 separate tests adopting our approach utilizing  $M = 100, 1000, 10000$  identical ANNs respectively have been carried out. As shown in Fig. 2 the robust estimate of  $\hat{p}_F$  obtained from the proposed approach converges to the true value (i.e. blue dashed horizontal line) when  $M = 1000$  identical ANNs are used. This means that  $M = 1000$  ANNs is sufficient enough to explore the entire solution space of the error function, thus locating a global minima. The importance of this approach is that it can lead to significant improvements in the predictions  $\hat{p}_F$ , while involving little additional computational effort.



(a) 5<sup>th</sup> and 95<sup>th</sup> Percentile Confidence Intervals

(b) Pdf Representing Model Uncertainty

Figure 2: Confidence Intervals and Probability Density Functions Representing Model Uncertainty for  $M = 100, 1000, 10000$  Identical Trained Artificial Neural Networks

### 4.4 CANTILEVER BEAM

A cantilever beam of length  $L$  and rectangular cross section of width  $b$  and height  $h$  is loaded at the end by a concentrated point load  $P$ . The displacement  $w$  at the tip of the beam should be

determined for the case where the point load  $P$ , the Young's modulus  $E$ , the density  $\rho$  of the material and the height  $h$  are uncertain.

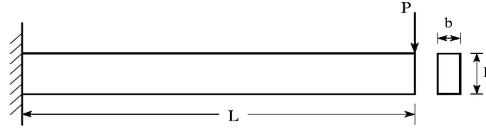


Figure 3: Cantilever Beam

Uncertainties of the width  $b$  and of the length  $L$  are assumed to be negligible. The displacement  $w$  at the tip of the beam where load  $P$  is applied can be expressed mathematically as:

$$w = \frac{\rho g b h L^4}{8EI} + \frac{PL^3}{3EI} \quad (19)$$

where  $g$  denotes gravitational constant, and  $I$  is given as:

$$I = \frac{bh^3}{12} \quad (20)$$

The limit state function for the cantilever beam is defined as:

$$G_{cantilever}(\mathbf{x}) = \beta - w(\mathbf{x}) \quad (21)$$

where  $\beta = 0.01$  is the maximum allowable displacement of the beam. In this example, the parameter uncertainties are modelled as random variables characterised by probability density function given in Table. 2.

Parameter	Distribution	$\mu$	$\sigma$	SI - unit
$P$	Log - Normal	5	0.4	KN
$h$	Normal	0.24	0.01	m
$\rho$	Log - Normal	600	140	Kg/m <sup>3</sup>
$E$	Log - Normal	10	1.6	GN/m <sup>2</sup>

Table 2: Model Input Parameters

The aim of this example is to study how small number of training samples (i.e.  $N_{train} = 50, 100, 150, 200$ ) affects the robust estimate and the corresponding confidence intervals.

#### 4.5 ANALYSIS 3

In this section, 2 sets (i.e., similar to Section 4.2) of identical ANNs (i.e.  $M = 1000$ ) with hidden layer configuration of (4,7,1) have been constructed based on  $D_{train}(\mathbf{x}, \mathbf{y}), N_{train} = 50, 100, 150, 200$  obtained via LHS algorithm [19]. The approach used in Section 4.2 has been adopted here to estimate the posterior probability and  $R^2$  value of the ANNs. Further, to accurately compute an estimate of  $\hat{p}_F$ ,  $10^4$  Monte Carlo simulation runs have been used for each ANN in the first set, and the proposed approach presented have been used to combine the prediction made by each ANN in the first set into a robust  $\hat{p}_F$  estimate. Finally, the model uncertainty has been

quantified in terms of confidence intervals of the robust estimate. The results of these analyses are shown in Fig. 4.

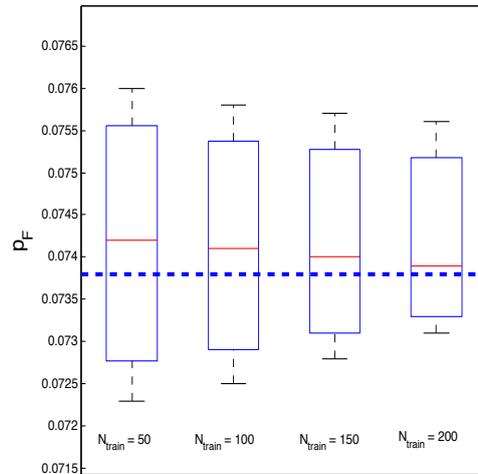


Figure 4: Robust Confidence Intervals Obtained from Different Number of Training Samples ( $N_{train} = 50, 100, 150, 200$ )

Notice that the "true" (i.e., reference) value of the failure probability (i.e.,  $\hat{p}_F = 0.0738$ , shown in blue dashed lines in Fig. 4) has been obtained with a large number samples (i.e.,  $N = 10^4$ ) of simulations of the original model to provide a robust term for comparison. Also, from the results in Fig. 4, as the number of training samples  $N_{train}$  increases, the width of the confidence intervals decreases and the expected value of the robust estimate approaches the reference value (i.e.,  $\hat{p}_F = 0.0738$ ). On the other hand, in the cases of small training data sets (e.g.,  $N_{train} = 50, 100$ ) the failure probabilities are significantly overestimated by the proposed approach (e.g., the expected values of the robust estimate are far off from the reference value) and the associated model uncertainties are quite large. However, in all the cases for small training data sets, the confidence intervals derived is robust enough to capture the true estimate. Hence, in a situation where training data set is small, this approach can be used as a guide to derive a robust confidence interval that is adequate to capture the true value that is being estimated.

## 5 CONCLUSIONS

Reliability analysis of complex models using the simulation approach is computationally expensive due to the large number of model evaluations required to compute their robust measures. In this paper, an ANN is being used as substitutes for an expensive model to alleviate the computational restrictions. The use of ANN for this kind of analysis introduces additional biasing and variance (i.e. uncertainties) to the predicted quantity. It has been shown that the use of cross-validation technique to select the best ANN out of a set of ANN with identical architecture introduces biasing and reduces the robustness of the predicted quantity. Therefore, a novel approach has been presented to enhance the accuracy of the prediction (i.e. robustness) made by an ANN and quantify the model uncertainties in terms of confidence intervals. The proposed approach combines Bayesian model selection and model averaging technique into a unified framework. The applicability of the proposed approach has been demonstrated on two examples. Although the computational effort required for implementing the proposed approach is expensive, parallelization strategies can be adopted to reduce this effort.

## Acknowledgements

This work has been partially supported by the EPSRC Grant EP/M018717/1 (Smart on-line monitoring for nuclear power plants (SMART)).

## REFERENCES

- [1] R. E. Melchers, *Structural reliability*, Horwood, 1987.
- [2] G. Taguchi, *Introduction to quality engineering: designing quality into products and processes*, 1986.
- [3] I. M. Sobol, Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates, *Mathematics and computers in simulation* 55 (1) (2001) 271–280.
- [4] E. Cashwell, C. Everett, *Monte-Carlo methods*, Pergamon, London, 1959.
- [5] R. Melchers, Importance sampling in structural systems, *Structural safety* 6 (1) (1989) 3–10.
- [6] O. Ditlevsen, P. Bjerager, R. Olesen, A. Hasofer, Directional simulation in gaussian processes, *Probabilistic Engineering Mechanics* 3 (4) (1988) 207–217.
- [7] M. de Angelis, E. Patelli, M. Beer, Advanced line sampling for efficient robust reliability analysis, *Structural Safety* 52 (2015) 170–182.
- [8] S.-K. Au, E. Patelli, Rare event simulation in finite-infinite dimensional space, *Reliability Engineering & System Safety* 148 (2016) 67–77.
- [9] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, Tech. rep., DTIC Document (1985).
- [10] U. Oparaji, R.-J. Sheu, M. Bankhead, J. Austin, E. Patelli, Robust artificial neural network for reliability and sensitivity analysis of complex non-linear systems, Submitted to Neural Networks.
- [11] T. Nilsen, T. Aven, Models and model uncertainty in the context of risk analysis, *Reliability Engineering & System Safety* 79 (3) (2003) 309–317.
- [12] E. Zio, A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes, *IEEE Transactions on Nuclear Science* 53 (3) (2006) 1460–1478.
- [13] M. Bayarri, J. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. Parthasarathy, R. Paulo, J. Sacks, D. Walsh, Computer model validation with functional output, *The Annals of Statistics* (2007) 1874–1906.
- [14] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, J. Tu, A framework for validation of computer models, *Technometrics*.
- [15] M. C. Kennedy, A. O’Hagan, Bayesian calibration of computer models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63 (3) (2001) 425–464.

- [16] D. G. Kleinbaum, M. Klein, Maximum likelihood techniques: An overview, in: Logistic regression, Springer, 2010, pp. 103–127.
- [17] T. H. Wonnacott, R. J. Wonnacott, Introductory statistics, Vol. 19690, Wiley New York, 1972.
- [18] B. Sudret, et al., Comparing probabilistic and p-box input modelling in structural reliability analysis.
- [19] J. C. Helton, F. J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Reliability Engineering & System Safety 81 (1) (2003) 23–69.