Performance Comparison between Multiple-Output Artificial Neural Networks and Classic Surrogate Models for System Reliability Problems

Henrique Machado Kroetz[1], Bruno Gustavo dos Santos[1]

[1]*Center for Marine Studies, Federal University of Paraná*
Av. Beira Mar, s/n, Paraná, Brazil
henrique.kroetz@ufpr.br, bruno.santos1@ufpr.br

Abstract. Artificial neural networks (ANNs) have been successfully used as a surrogate model in structural reliability analysis due to their ability to model complex, nonlinear relationships between input and output variables. In this context, the ANN model is usually trained using input variables such as material and geometrical properties to learn the relationship between the inputs and the output variable, which is the probability of failure. Despite their potential, ANNs are often overlooked in favor of more robust and easier-to-train models, such as Polynomial Chaos Expansions and Kriging. However, in system reliability problems with multiple outputs, ANNs offer an advantage as they can handle multiple outputs in their default formulations, avoiding the need for multiple surrogates or complex formulations, thus reducing computational costs. This paper aims to compare the performance of ANNs with other surrogate models in this context. Two examples are addressed comparing ANNs, Kriging and Polynomial Chaos Expansions surrogate models. Results suggest that using multiple-output ANNs for surrogating all limit states at once is more efficient than training separate networks for each limit state, but more studies are required in order to propose a comprehensive strategy.

Keywords: Structural Reliability, System Reliability, Artificial Neural Network

1 Introduction

Artificial Neural Networks (ANNs) are a class of machine learning models inspired by the structure and function of the human brain. Their capabilities of capturing complex nonlinear relationships and handling high-dimensional datasets have rendered them useful for many different application on several fields. In the context of reliability analysis, ANNs are often employed as surrogate models to emulate computationally burdensome limit state equations [1–3]. As analytical models, ANNs are usually much faster to evaluate, so that Monte Carlo simulations can be performed to estimate probabilities of failure. One of the key advantages of ANNs lies in their ability to capture complex dependencies among input variables without relying on assumptions of linearity or distributional properties. Unlike other commonly-used surrogate models (e.g. Kriging, polynomial response surfaces), which typically assumes a smooth, continuous response surface, ANNs can effectively capture nonlinearity, discontinuity, and interactions among variables, potentially leading to more accurate and robust reliability predictions [4]. This paper aims to examine the behavior of straight-forwardly built artificial neural networks in comparison to polinomial chaos expansions (PCE) and Kriging for addressing system-reliability problems. All such metamodels have been considerably developed in the last few years, with several variations to the formulations of each being proposed. In this work, basic formulations are considered for all surrogate models, following architectures more commonly used in structural reliability analysis.

Even though ANNs have been shown to be an effective tool for reliability analysis, competing surrogate modeling techniques are more commonly used in this context. ANNs typically require a significant amount of training data and computationally intensive calibrating procedures, including optimization algorithms and hyperparameter tuning. Both other models adressed in this paper, Kriging and polynomial chaos expansions, assume that the underlying response surface is smooth and continuous. This assumption aligns well with many engineering systems. In contrast, ANNs are known for their ability to capture nonlinearity and discontinuity, which may not always be necessary or desired in certain structural reliability analysis applications. On the other hand, most surrogate mod-

els employ a single-output regression logic in their formulation, in such a way that dealing with multiple output is either not possible or requires a much more complex formulation. As a result, multiple-output problems are usually addressed by the construction of more than one surrogate model when such techniques are applied, which obviously lead to increased computational burden. This might be a problem when approaching system-reliability problems. Simple series-systems can be addressed by building a single surrogate model to emulate the minimum value between all limit-state equations, but the original model being surrogated tends to present discontinuities which can jeopardize representation quality in inherently continuous surrogate models. More complex systems may require every single (potentially burdensome) limit state to be evaluated multiple times. ANNs have an advantage on this regard, since considering multiple outputs can be as simple as adding more neurons to the network's output layer. Thus, they might be a better tool for handling system-reliability problems. Some formulations have been proposed specifically in order to address system-probability problems using Kriging and PCE, but approaching the system as a single-limit state with discontinuities [5–8]. In this context, the present study evaluates the performance of PCE, Kriging and ANN on surrogating limit state equations in system-reliability problems.

2 Basics of surrogate models

Surrogate models or metamodels are analytical approximations that simulate the behavior of more complex models. By leveraging knowledge of the outputs of the original model (e.g., a numerical model), for certain sets of input data, the metamodel is adjusted and can be analyzed as a substitute for the original model. The input data sets used for this purpose are referred to as "support points", and the set of all carefully selected support points is called a design of experiment (DOE). Effective metamodels provide a sufficiently close response to that which would be obtained by evaluating the model being represented, referred herein as "high-fidelty model". Let the high-fidelty model $\mathcal{M}$ be a "black box" model, meaning only the relationship between input and output data of the model is considered. Based on these relationships, a metamodel $\tilde{\mathcal{M}}$ is defined as shown in Equation 1:

$$\tilde{\mathcal{M}}(\boldsymbol{x}) \approx \mathcal{M}(\boldsymbol{x}) = \boldsymbol{y}, \tag{1}$$

where $\boldsymbol{y}$ gathers the high fidelity model responses. Notice that the output is a vector, meaning a single surrogate model may have multiple scalar outputs. By leveraging statistical techniques, surrogate models construct simplified analytical expressions or numerical mappings that encapsulate the underlying system behavior. These surrogate models effectively serve as surrogates for the computationally expensive simulations, enabling rapid and accurate estimations of system responses over a wide range of input conditions. Several different types of functions and training strategies can be employed in order to build the surrogate models. In this work, a few of such strategies will be compared. They are briefly explained in Sections 2.1 to 2.3.

2.1 Kriging

Gaussian process regression or Kriging is a metamodel that assumes the computational model of interest can be approximated by an underlying Gaussian stochastic process realization [9, 10], as shown in Equation 2.

$$\mathcal{M}^{krig}(\mathbf{x}) = \sum_{j=1}^{p} \beta_j^{krig} f_j^{krig}(\mathbf{x}) + Z(\mathbf{x}), \tag{2}$$

where the first term is a deterministic trend and $Z(\mathbf{x})$ is a Gaussian process of zero mean. Once again, training considers a DOE $\mathcal{X} = \{\boldsymbol{\chi}^{(i)}, i = 1, \ldots, n\}$ associated to high-fidelty model responses $\boldsymbol{\mathcal{Y}} = \{\mathcal{Y}^{(i)} = \mathcal{M}(\boldsymbol{\chi}^{(i)}), i = 1, \ldots, n\}$. Equation 3 shows the least-squares approach employed in order to estimate the parameters:

$$\widehat{\boldsymbol{\beta}}^{krig}(\boldsymbol{\theta}) = \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{\mathcal{Y}}, \tag{3}$$

where $\boldsymbol{F}$ is a matrix that gathers the regression functions evaluated at the DOE points., i.e. $F_{ij} = f_j^{krig}(\boldsymbol{\chi}^{(i)})$ and $\boldsymbol{R}$ is the autocorrelation matrix where $R_{ij} = R(\boldsymbol{\chi^{(i)}}, \boldsymbol{\chi^{(j)}}; \boldsymbol{\theta})$. The variance of the process is estimated as shown in Equation 4

$$\widehat{\sigma}^2(\boldsymbol{\theta}) = \frac{1}{m}(\mathcal{Y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}^{krig})^T \boldsymbol{R}^{-1} (\mathcal{Y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}^{krig}). \tag{4}$$

Vector $\boldsymbol{\theta}$ gathers correlation hypeparameters, which are determined using a basic maximum likelyhood approach. Once the model is properly trained, the prediction for new points $\boldsymbol{x}$ follows a normal distribution $\widetilde{\mathcal{M}}^{krig} \sim \mathcal{N}(\mu_{\widetilde{\mathcal{M}}}(\boldsymbol{x}), \sigma^2_{\widetilde{\mathcal{M}}}(\boldsymbol{x}))$, with mean and variance given by Equations 5 and 6, respectivelly.

$$\mu_{\widetilde{\mathcal{M}}}(\boldsymbol{x}) = f^T(\boldsymbol{x})\boldsymbol{\beta} + \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{R}^{-1}\big(\boldsymbol{y} - \boldsymbol{F}^T\boldsymbol{\beta}\big), \tag{5}$$

$$\sigma^2_{\widetilde{\mathcal{M}}}(\boldsymbol{x}) = \sigma^2\Big(1 - \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{r}(x) + \boldsymbol{u}^T(\boldsymbol{x})\big(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}\big)^{-1}\boldsymbol{u}(\boldsymbol{x})\Big), \tag{6}$$

with $\boldsymbol{r}(\boldsymbol{x}) = [R(\boldsymbol{x}, \boldsymbol{\chi}^{(1)}), \ldots, R(\boldsymbol{x}, \boldsymbol{\chi}^{(n)})]$ and $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x})$. Hyperparameters have to be tuned in order for the Kriging model to present precise results. In this work, a maximum likelihood approach is employed in this step.

2.2 Polynomial Chaos Expansion

Polynomial Chaos Expansions (PCE) have been utilized since the 1990s to solve problems involving stochastic finite elements, and became a popular tool to be used as surrogate models for structural reliability problems in the last 20 years. Through PCE, the response $\tilde{Y}$ of a structural system is treated as a random variable belonging to a specific space, such as the space of random variables with finite variance [11]. Thus, it can be represented as a linear combination of the vectors in a basis of this particular space:

$$\mathcal{M}^{pce} = \tilde{Y} = \sum_{\alpha} \Psi_\alpha a_\alpha, \tag{7}$$

where Ψ_α are the multivariate polynomial basis functions, that are orthonormal with respect to the densities $f_{\mathbf{X}}(x)$:

$$E\big(\Psi_i(\mathbf{X}), \Psi_j(\mathbf{X})\big) = \delta_{ij}, \tag{8}$$

The coefficients a_α are to be determined. For practical computational reasons, the series must be truncated to a certain degree. The truncated PCE representation is directly proportional to the truncating order, which determines the complexity of the model. A low-complexity model may not be sufficient to represent the behavior of the system, while a high-complexity model may lead to overfitting. A representative basis can be constructed through tensorization of the truncated univariate expansions of the input random variables. The number of basis functions selected n_b can be determined by a function of the number of random variables m and the maximum degree to be adopted p, as shown in Equation 9

$$n_b = \frac{(m+p)!}{m!p!}, \tag{9}$$

A practical algorithm to build the multivariate basis is presented in Sudret et al. (2006). The a_α may be obtained non-intrusively by least squares regression:

$$\mathbf{A} = (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\mathbf{\Psi}^T Y, \tag{10}$$

where $\mathbf{A}$ is a vector that gathers the coefficients a_α and $\mathbf{\Psi}$ is the data matrix where $\Psi_{ij} = \Psi_j(\mathbf{X}^{(i)})$.

2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the brain's neural networks. ANNs consist of interconnected nodes or "neurons" that process input data, learn patterns, and make predictions. Through weight adjustments and non-linear transformations, ANNs can capture complex relationships in data. Their versatility has led to applications in several fields, including its usage as surrogate models in reliability analysis. This works focuses on the most popular and frequently used kind of ANN, the Multilayer Perceptron (MLP). A multilayer perceptron is a type of artificial neural network that consists of multiple layers of interconnected artificial neurons or nodes. It is a feedforward neural network, meaning the information flows in one direction, from the input layer through the hidden layers to the output layer. Each neuron in the MLP receives inputs, applies a weighted sum of the inputs along with a bias term, and applies an activation function to produce an output. The hidden layers between the input and output layers allow the MLP to learn complex nonlinear relationships and capture higher-level abstractions in the data. MLPs are often used for tasks such as classification, regression, and pattern recognition, and they can be trained using techniques like backpropagation to adjust the weights and biases to minimize the error between the predicted and target outputs. The structure of the MLP depends on the considered optimization algorithm and loss function. In this work, the ADAM algorithm [12, 13] was adopted to minimize error in simple MLP neural networks, but other optimization algorithms could be employed on this step.

Following these guidelines, an ANN surrogate models with 3 layers with M neurons on the first layer, $h^{(rn)}$ on the second, and one on the last, would approximate the response following Equation 11:

$$\tilde{\mathcal{M}}^{(rn)}(\boldsymbol{x}) = \sum_{j=1}^{h} w_{j13}^{(rn)} f_a\big(a_{j2}^{(rn)}\big), \tag{11}$$

where w are the weights and f_a are the activation functions.

3 Surrogate Models in System Reliability Problems

Let X be a vector that gathers the random variables of a problem, with joint probability density function $f_{\boldsymbol{X}}(\boldsymbol{x})$. Let $\mathcal{F}$ be the event where the studied system fails (*i.e.* a failure associated with the problem). The probability of failure associated with a given limit state i is given by:

$$P_f = \int_{D_{fi}} f_{\boldsymbol{X}}(\boldsymbol{x}) \boldsymbol{dx}, \tag{12}$$

where D_{fi} is the i-th failure domain, defined as the set of all points where the corresponding limit state function assumes negative values. In general, solving this integral is challenging, and numerical methods such as Monte Carlo simulations are employed for its resolution. The computational cost of this approach can be high, especially in cases involving multiple random variables, time dependency or numerical models with numerous degrees of freedom [14–16]. Therefore, it is common to replace limit state functions with metamodels in reliability problem solving, thereby enabling computationally feasible solutions.

Typically, multiple limit states can be considered simultaneously, as real engineering systems are often associated with multiple failure modes. A system-failure domain composed of different limit states is often bounded by a non-differentiable failure surface at points where individual failure domains intersect. This can pose challenges for certain solution strategies, particularly when dealing with metamodels whose approximating function is smooth. Furthermore, the systems leading to structural failure can exhibit various organizational arrangements. The most common are series systems, where the violation of any of the nls associated limit states results in failure ($\mathcal{F} = \bigcup_{i=1}^{nls} \mathcal{F}_i$), and parallel systems, where the violation of all nls limit states is necessary for failure to occur ($\mathcal{F} = \bigcap_{i=1}^{nls} \mathcal{F}_i$). Additionally, mixed systems may depend on specific combinations of limit states. When surrogate models are applied in this context, one can choose to surrogate each limit state individually, composing the system solution based on the individual approximations, or to surrogate the system response directly. The first approach leads to de construction of several surrogate models, which could render inefficient. The latter, leads to the construction of a single surrogate model that approximates a complex region, potentially with discontinuities and non-derivable points. Since ANNs are the only surrogate models considered in this paper which are not impaired by the presence of such difficulties, they are tested in Section 4 for system reliability problems, where their performance is compared with that of more classical approaches.

4 Performance Comparison

To compare the performance of the different metamodeling techniques in system reliability problems, similar parameters will be adopted for all the tests. While this does not necessarily imply that all trainings are conducted in the same manner, we prioritize consistency in computational cost. Thus, comparable total quantities of support points will be used for each analysis. This takes into account that Artificial Neural Networks (ANNs) need to divide the total points between training and validation sets, and allocate the total available points for a given analysis among the different quantities of metamodels that need to be built for individual limit states. Furthermore, in order to maintain some level of generality in the conclusions, the simplest and most widespread available approaches will be adopted for training. The support points used will be the same for all metamodels in each analysis, and Latin Hypercube Sampling will be employed on all cases. The number of support points will be increased for each surrogate model until the obtained failure probability is close enough to the reference value. A 1% error is adopted as stopping criterion. Additionally, since adaptive strategies vary a lot between different types of surrogate models, all training will be static (i.e., adaptive schemes will not be employed). Since polynomial response surface's showed by far the worst performance, its results will be omitted in order to focus on more relevant results. This was expected, since they are comparably very simple models, with lower generalization capabilities [17]. All Kriging

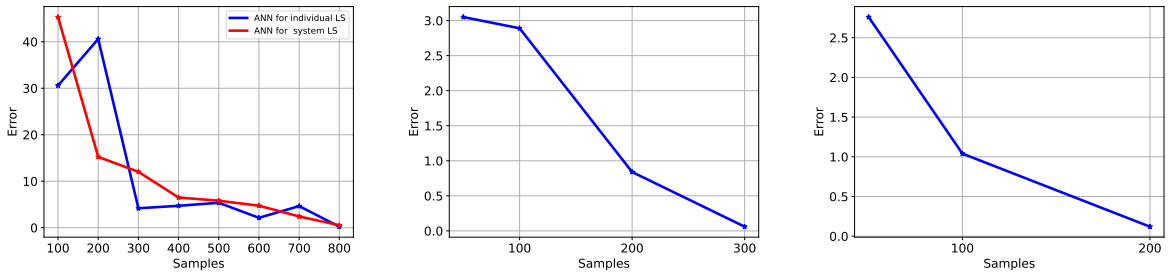
metamodels used a linear trend, and hyperparameters were selected as detailed in Section 2.1. For PCE, the usual formulation was employed, so that truncation to a maximum degree of 3 was carried out as discussed in Section 2.2. For the ANN, two different cases were considered on each example. First, a single MLP was trained for each individual limit state, following the architecture of Kolmogorov–Nielsen theorem. A second, more complex neural network was built in order to represent all limit states in a single surrogate model. This network's architecture is defined as one neuron for each input on the first layer, one neuron to each output on the last layer, and three processing layers with decreasing number of neurons. In this case, system reliability is explicitly calculated by composing the multiple outputs of the network. In all other cases, the system's response is composed regarding all de individual limit state surrogate models. Two examples are addressed herein, representing the two most common system reliability configurations.

4.1 Parallel system benchmark problem

This benchmark example was first presented in [18], and consists in four limit states composing a parallel system, as shown in Equation 13. All five independent random variables follow standard normal distributions.

$$g_{sys}(\mathbf{X}) = \max\Big[\big(2.677 - X_1 - X_2\big); \big(2.500 - X_2 - X_3\big); \big(2.323 - X_3 - X_4\big); \big(2.250 - X_4 - X_5\big)\Big] \quad (13)$$

The reference result $P_{f_{sys}} = 2.13 \ . \ 10^{-4}$ was obtained via Monte Carlo simulation associated with LHS sampling, using 10^6 samples. DOE of increasing sizes were considered for all surrogate models, and each configuration was run 10 times. Results are shown in Figures 1a, 1b and 1c, representing the average error of the 10 runs for each DOE of each metamodel technique for $P_{f_{sys}}$.



(a) ANN (b) PCE (c) Kriging

Figure 1. System P_f error for ANN, PCE and Kriging surrogates

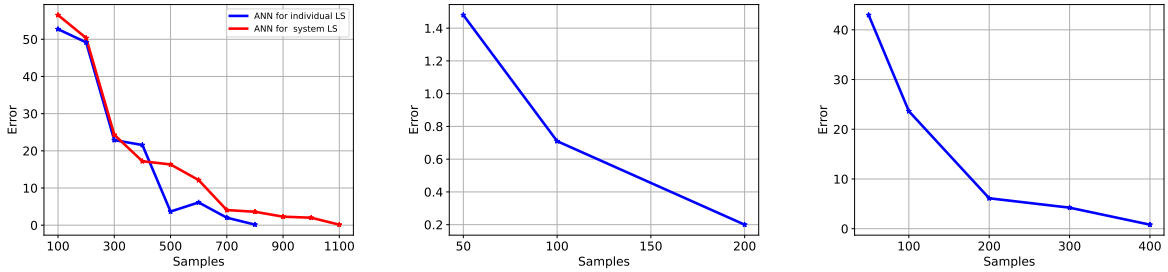
In this example, both ANNs approaches showed a similar behavior, with very precise results achieved in some runs of each DOE configuration and poor results achieved in others. Hence, the average result took a while to converge to the reference, with both individual and system ANN converging at similar rates. The system networks training, however, never took as long as the time to train two individual limit state neural networks. Hence, it was significantly faster to train when compared to the training time needed for all 4 individual networks. Even so, PCE and Kriging surrogate models in their common simple configurations showed a significantly better performance than both ANN metamodels.

4.2 Series system benchmark problem

This example was first presented by [19], and consists in the series system of nonlinear equations represented in Equation 14, where all random varibles follow standard normal distributions.

$$g_{sys}(\mathbf{X}) = \min\Big[\big(2 - X_2 + e^{-0.1X_1^2} + (0.2X_1)^4\big); \big(4.5 - X_1X_2\big)\Big] \quad (14)$$

Results are shown in Figures 2a, 2b and 2c. The reference result $P_{f_{sys}} = 3.47 \ . \ 10^{-3}$ was obtained via brute Monte Carlo simulation, using 10^6 samples. Once again, DOE of increasing sizes were considered for all surrogate models, and each configuration was run 10 times.



(a) ANNs (b) PCE (c) Kriging

Figure 2. System P_f error for different surrogate models

Both ANN showed a similar behaviour with smaller DOE, again with very precise results being obtained inconsistently. The individual ANNs approach converged faster to the solution, with a large number of support points being necessary for the multiple-output network to present consistently precise results. On the other hand, however the system neural networks took longer to train, training time was comparable to that of each individual neural networks on all cases. This result is consistent with that obtained in the series-system example, showing an advantage in adopting the approach investigated in this paper when it comes to employing ANNs as surrogate models. Once again, the other classical metamodel techniques showed better performane when compared to both ANN approaches.

5 Conclusions

This paper studied the performance of multiple-output aritifical neural networks on system-reliability problems, in comparison with Polynomial Chaos Expansion, Kriging and ANNs built only for individual limit states. One series and one paralel system benchmark examples were considered for all four approaches. Increasing number of points were considered for static DOEs, so that the performance between different surrogate models could be compared when the same amount of data is available. On all cases, the multiple output ANN showed a similar or worse precision in comparison with all other surrogate models. On the other hand, both ANN approaches were the ones to achieve the best results for small amounts of support points, but inconsistently, so that other runs would have such bad results that the averages were quite bad. Interestingly, the multiple-outputs ANNs took considerably less time to train than all the individual networks required for each problem took together. This is a significant advantage in the context of surrogate modeling.

In this work, the formulations of all metamodels were kept as simple as possible, so as to compare widely used techniques without too many specificities. Nevertheless, ANNs are known for having a huge number of possible formulations, adapting very well to the characteristics of each problem being addressed. Hence, it can be concluded that despite the performance obtained in this study being worse than that of PCE and Kriging, it is worth investigating the performance of multiple-output networks whose formulation is competitive for specific problems. This will be considered in further studies by the authors.

Acknowledgements. The authors thank the National Council for Scientific and Technological Development - CNPq for the financial support.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] A. Hosni Elhewy, E. Mesbahi, and Y. Pu. Reliability analysis of structures using neural network method. *Probabilistic Engineering Mechanics*, vol. 21, n. 1, pp. 44–53, 2006.

[2] H. M. Kroetz, R. K. Tessari, and A. T. Beck. Performance of global metamodeling techniques in solution of structural reliability problems. *Advances in Engineering Software*, vol. 114, pp. 394–404, 2017.

[3] de W. J. Santana Gomes. Structural Reliability Analysis Using Adaptive Artificial Neural Networks. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, vol. 5, n. 4, 2019.
[4] C. C. Aggarwal". *"Neural Networks and Deep Learning - A Textbook"*. "Springer Cham", 2018.
[5] J. Zhang, H. W. Huang, and K. K. Phoon. Application of the kriging-based response surface method to the system reliability of soil slopes. *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 139, n. 4, pp. 651–655, 2013.
[6] X. Yang, Y. Liu, C. Mi, and C. Tang. System reliability analysis through active learning kriging model with truncated candidate region. *Reliability Engineering & System Safety*, vol. 169, pp. 235–241, 2018.
[7] J. Wang, Z. Sun, and R. Cao. An efficient and robust kriging-based method for system reliability analysis. *Reliability Engineering & System Safety*, vol. 216, pp. 107953, 2021.
[8] N.-C. Xiao, K. Yuan, and H. Zhan. System reliability analysis based on dependent kriging predictions and parallel learning strategy. *Reliability Engineering & System Safety*, vol. 218, pp. 108083, 2022.
[9] B. Ankenman, B. L. Nelson, and J. Staum. Stochastic kriging for simulation metamodeling. *Operations research*, vol. 58, n. 2, pp. 371–382, 2010.
[10] B. Echard, N. Gayton, and M. Lemaire. Ak-mcs: An active learning reliability method combining kriging and monte carlo simulation. *Structural Safety*, vol. 33, n. 2, pp. 145–154, 2011.
[11] H. M. Panayirci and G. I. Schuëller. On the capabilities of the polynomial chaos expansion method within sfe analysis—an overview. *Archives of Computational Methods in Engineering*, vol. 18, n. 1, pp. 43–55, 2011.
[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, vol. , 2014.
[13] A. G. Carlon, H. M. Kroetz, A. J. Torii, R. H. Lopez, and L. F. F. Miguel. Risk optimization using the chernoff bound and stochastic gradient descent. *Reliability Engineering & System Safety*, vol. 223, pp. 108512, 2022.
[14] G. P. Pellizzer, H. M. Kroetz, E. D. Leonel, and A. T. Beck. Time-dependent reliability of reinforced concrete considering chloride penetration via boundary element method. *Latin American Journal of Solids and Structures*, vol. 17, n. 8, 2020.
[15] H. Kroetz, M. Moustapha, A. Beck, and B. Sudret. A two-level kriging-based approach with active learning for solving time-variant risk optimization problems. *Reliability Engineering & System Safety*, vol. 203, pp. 107033, 2020.
[16] R. K. Tessari, H. M. Kroetz, and A. T. Beck. Performance-based design of steel towers subject to wind action. *Engineering Structures*, vol. 143, pp. 549–557, 2017.
[17] M. Rashki, H. Azarkish, M. Rostamian, and A. Bahrpeyma. Classification correction of polynomial response surface methods for accurate reliability estimation. *Structural Safety*, vol. 81, pp. 101869, 2019.
[18] P. Bjerager. Probability integration by directional simulation. *Journal of Engineering Mechanics*, vol. 114, n. 8, pp. 1285–1302, 1988.
[19] F. Grooteman. An adaptive directional importance sampling method for structural reliability. *Probabilistic Engineering Mechanics*, vol. 26, n. 2, pp. 134–141, 2011.