COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING
ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-0789
ESTIMATION OF POLLUTION SOURCES WITH PHYSICS-INFORMED AND BAYESIAN NEURAL NETWORKS

Roberto Mamud
Universidade Federal do Rio de Janeiro - Campus Macaé. Av. Aluizio da Silva Gomes, 50, Novo Cavaleiros, Macaé - RJ, 27930-560
rmamud@macae.ufrj.br

Carlos T. P. Zanini
Universidade Federal do Rio de Janeiro. Av. Athos da Silveira Ramos, 149 - Edifício do Centro de Tecnologia, Bloco C (Térreo), Cidade Universitária - RJ, 21941-909
carloszanini@dme.ufrj.br

Helio S. Migon
Universidade do Estado do Rio de Janeiro. Rua Bonfim, 25, Vila Amélia, Nova Friburgo - RJ, 28625-570
migon@im.ufrj.br

Antonio J. Silva Neto
Universidade Estadual do Rio de Janeiro. Rua Bonfim, 25, Vila Amélia, Nova Friburgo - RJ, 28625-570
ajsneto@iprj.uerj.br

Abstract. *The use of the inverse problem approach in the identification of pollution sources in the environment has received an increasing attention in recent years. In this work, the inverse problem of identification of pollution sources location and intensity in a river is studied considering the advection-diffusion-reaction equation for the pollutant concentration, along with a Neural Network approach. In the forward problem, the location and intensity of the source term are known, as well as the other parameters present in the modelling of the phenomena involved. Additionally, because no pollutant was taken into account at the beginning, the initial condition is considered null, and the boundary conditions are set by the pollutant concentration on the boundary of the region of interest. In this way, the forward problem is solved by classical numerical methods. Such a numerical solution is then used as an input dataset for the inverse problem formulation and solution. The inverse problem is solved both by a Physics-Informed Neural Network (PINN), and by a Bayesian Neural Network (BNN). The PINN is a recent type of neural network that is trained to satisfy also the physical laws that describe the phenomena involved. However, it does not consider the possible uncertainties that are intrinsic to real world applications, making it natural to consider a Bayesian approach. In the Bayesian framework, these neural networks (PINN and BNN) serve as priors and such an approach yields a full posterior distribution of the parameters of interest. Therefore, in this work we show, with numerical examples, that the BNN can be a better option than the PINN alone, when dealing with noisy measurements. Numerical experiments related with PINN and BNN approaches are presented, demonstrating the feasibility of the strategy considered.*

Keywords: *Inverse source problems, Machine Learning, Physics-Informed Neural Networks, Bayesian Neural Networks*

1 INTRODUCTION

Over the last centuries, with the development of industrial production, which started with the industrial revolution, the world was capable of producing in larger quantities, improving the quality process, and reducing the time of goods production. However, with this increasing industrial process, the pollution also increased, generating, in many cases, contaminants in the air, soil or water. Unfortunately, the leakage of pollutant into water bodies like rivers, lakes or ocean is a common fact nowadays. This kind of pollution source includes the industrial waste, oil refineries, and wastewater treatment facilities. Here, we consider the model posed by Advection-dispersion-reaction equation, modelling the concentration of a pollutant in a river.

In this context, the area of Inverse Problems rises as an ally in the study of these leakages. For example, when it is observed that there is a pollutant in the riverside region, it is important to know where this leakage came from. In the Inverse Problem framework, we say that from boundary measurements, we are interested in determining the location of the pollution source, (El Badia *et al.*, 2005) and (Andrle and Badia, 2015). This kind of inverse problem is posed in contrast with the so-called direct problem, (Moura Neto and Silva Neto, 2013). In the example above, the direct problem can be formulated as follows: given the parameters of the medium and the location of the source pollution term, we are interested in the estimation of pollution concentration on the riverside region.

The direct problem is a well-posed problem in the Hadamard sense (Hadamard, 1902), while the inverse problem is usually ill-posed in view of the lack of uniqueness (Isakov, 1990). To contour this problem, it is common suppose that the source that we expect to reconstruct is in a specific class of function. Beside this, the numerical methods employed in each formulation (direct and inverse) must be different to avoid the so-called Inverse Crime; that is, according to (Colton and Kress, 2013) it is crucial that the synthetic data be obtained by a forward solver which has no connection to the inverse solver under consideration.

In this work, we employ classical numerical methods to solve the forward problem and a special type of Neural Network to solve the inverse source problem. The special types considered in this work are the Physics-Informed Neural Network (PINN) and the Bayesian Neural Network (BNN).

The PINN is a recent type of neural network introduced in (Raissi *et al.*, 2019). These nets have the characteristic of taking into account the physical laws that describe the phenomena involved while, at the same time, training to solve some learning tasks. It can be made by modifying the Mean Squared Error Loss (the function where we are interested in minimizing to obtain the weights and biases for neural network). This modification is made by adding a summation representing the physical laws of the model with linear or non-linear PDEs, with the derivatives of the network using Automatic Differentiation, (Baydin *et al.*, 2018). In other words, the PINN can be derived by applying the chain rule for differentiating compositions of functions using automatic differentiation where it has the same parameters as the network representing the function solution, where the shared parameters can be learned by minimizing the mean squared error loss (Raissi *et al.*, 2019).

Despite the fact that this kind of neural network is known to be faster and more accurate than an usual artificial neural networks, it does not consider the possible uncertainties that are intrinsic to real-world applications. In any measurements computed, there is an inherent error, and that should be considered. The Bayesian Inference can then be used to introduce this feature in the network.

In this work we study the concentration of a pollutant in a river governed by the advection-dispersion-reaction equation. The forward problem is studied through classical numerical methods with the goal of generating a dataset to be used in the inverse problem framework with neural networks considering the equation (PINN) and how the uncertainty measurements propagate in the problem (Bayesian approach).

In Section 2, it is presented a general setting for the pollutant dispersion in the bi-dimensional case. Beside this, it is also presented the uni-dimensional direct and inverse problems considered for the pollutant source concentration. The Sections 3 and 4 are devoted to the PINN and the BNN approaches for the inverse problem, respectively. In Section 5 , numerical experiments are presented related with the identification of the source location term with the PINN. Partial conclusions and extensions of the work are presented in Section 6.

2 The Direct and Inverse Problems considered

In this section, we establish the forward and inverse problem formulations for the pollution source concentration. Consider the following one-dimensional problem

$$\begin{cases} \partial_t c - D\partial^2_{xx}c + V\partial_x c + Rc = F(x,t), \text{ for } x \in (0,L), t \in (0,T) \\ c(x,0) = g_0(x), \text{ for } x \in [0,L] \\ c(0,t) = g_l, \text{ for } t \in [0,T] \\ c(L,t) = g_r, \text{ for } t \in [0,T], \end{cases} \tag{1}$$

where $\partial_t c$ and $\partial_x c$ (and $\partial^2_{xx}c$) represent the partial derivative of concentration c [g/m^3] with respect to time, t [s], and space x [m], respectively. Besides, D [m^2/s] is the dispersion coefficient, V [m/s] is the velocity coefficient, R [1/s] is the reaction coefficient, with respect to time, Δc represents the Laplacian of function c and ∇c represents its gradient vector. F [g/m^{3}s] is the source term, g_0 is the initial concentration and g_b is the known concentration values at the boundary $\partial\Omega$. This problem has a unique solution, provided that u, F, g_0 and g belong to appropriated function spaces, see for example (Evans, 1998). Note that in the above problem, we consider constant boundary concentrations $g_l \geq 0$ and $g_r \geq 0$, but it could be considered transient concentrations that change along the time.

The Direct Problem for (1) has the aim of, given the parameters of the medium (coefficients of the PDE), the initial and boundary conditions and the source term, to find the concentration $c(x,t)$ in any point of the river, at any time. On the other hand, an Inverse Problem for this general setting is, given some measurements of pollutant concentration that can be obtained by sensors experimental or synthetically, to find the source term F and the concentration c that solve the problem (1).

It is well known in the literature that this general setting is a ill-posed problem, because of the lack of uniqueness, see for example (Isakov, 1990), (Kirsch, 2011) and (Tarantola, 2005). In view of this, we must assume that the source term F, that we expect to reconstruct, belongs to a special class of function. In this work, we suppose the source term, in Eq. (1), is given by

$$F := F_\lambda(x,t) = \lambda_i \left(e^{-(x-\lambda_l)^2}\right)\left(e^{-(t-\lambda_t)^2}\right), \text{ for } x \in (0,L), t \in (0,T), \tag{2}$$

where $\lambda = (\lambda_i, \lambda_l, \lambda_t)$, $\lambda_i > 0$ is the intensity of the source pollutant concentration, $\lambda_l \in [0, L]$ is the source location and $\lambda_t \in [0, T]$ is the time of maximum leakage. This kind of source represents a continuous punctual leakage scenario concentrated in position λ_l, with maximum value at time λ_t and decreasing intensity for $t > \lambda_t$.

3 The Physics-Informed Neural Network Approach

As mentioned in the Introduction, the recently developed Physics-Informed Neural Network (PINN) defines an optimization process, in which the weights and biases minimizes a special modified loss function that takes the physical laws relevant to the problem in consideration.

In this section, we present the main ideas about the PINN, and how we use them to solve the inverse problem of source estimation.

First, we point out the main differences between construct an usual Artificial Neural Network (ANN) and a Physics-Informed Neural Network.

The ANN is composed by a collection of neurons (or nodes) distributed in layers that apply different transformations on their inputs. This neurons are weighted to adjust the training process. The aim in the training phase of an ANN is to find a set of optimal weights that minimizes the error between the prediction (output) and the target values.

The main difference between the ANN and the PINN are the target values. In the ANN, we provide a list of target values that are used for tunning the parameters of the net in the loss function of Mean Squared Error type. On the other hand, in the PINN, there are no target values because we are interested in minimizing a new loss function that incorporates the physics of the problem through the differential equation that govern the observed data Raissi *et al.* (2019).

Consider the problem in the pollutant dispersion in a river, posed by Eq. (1). In the inverse problem, which is correlated to a corresponding observational data from sensors, there is a (unknown) source of type (2), depending on specific $\lambda_s = (\lambda_i^s, \lambda_l^s, \lambda_t^s)$, that we are interested in estimating.

Let us take into account an ANN whose inputs are the nodes $z_i^s = (x_i^s, t_i^s)$, for $i = 1, 2, \ldots, N_s$, where N_s correspond to the number of domain points that generate the sensor measurements $c_s(z_i^s)$, for $i = 1, 2, \ldots, N_s$. Note that the concentration vector from sensors $(c_s(z^s))$ is correlated to a (unknown) value of λ_s vector, to be determined. In view of simplify the notation, we will omit this dependency in the writing of $c_s(z_i^s)$. The output of this ANN is the function $c_{NN}(z_i^s, \theta)$, where θ is the vector of all parameters of the network.

Therefore, the loss function, $L_d(\theta)$, of Mean Squared Error (MSE) type for the ANN, uses only the data from the sensors and is given by

$$L_d(\theta) = \frac{1}{N_s}\sum_{i=1}^{N_s} |c_{NN}(z_i^s, \theta) - c_s(z_i^s)|^2. \tag{3}$$

For the purpose of defining the new loss function for the PINN case, we need to define the following functional, namely *residue* of Eq. (1):

$$\mathcal{R}(u, \lambda) := \partial_t u - D\partial_{xx} u + V\partial_x u + Ru - F_\lambda(x,t), \tag{4}$$

where u is an appropriated function and $F_\lambda(x,t)$ is given by (2). In a general way, we note that if $\mathcal{R}(c) = 0$, then we say that $u = c$ is a solution (general) of the PDE (1), with known parameters values D, V, R and λ.

Since we are approximating the pollutant concentration solution by a neural network $c_{NN}(z, \theta)$, then it will result in a different approximation by a neural network for $\mathcal{R}$, namely a Physics-Informed Neural Network (PINN) (Raissi *et al.*, 2019). This method involves training a neural network to approximate the concentration data in Eq.(3), and, simultaneously, the solution of the PDE in Eq. (1) by minimizing a new loss function that incorporates the residue given by Eq.(4) in a new summation term.

It is important to point out that we are assuming that as in the neural networks' output c_{NN} and as in the residue $\mathcal{R}(c_{NN}, \lambda)$, the set of parameters θ is the same. So, the parameters (θ, λ) can be optimized by minimizing the new loss function, $L(\theta, \lambda)$, defined by $L(\theta, \lambda) = L_d(\theta) + L_r(\theta, \lambda)$, where $L_d(\theta)$ is the loss term due to the data, given by Eq.(3), and $L_r(\theta, \lambda)$ is the loss term due to the residue, defined by

$$L_r(\theta, \lambda) = \frac{1}{N_r}\sum_{i=1}^{N_r} |\mathcal{R}(c_{NN}(z_i^r, \theta), \lambda)|^2, \tag{5}$$

with $\mathcal{R}(c_{NN}(z_i^r, \theta), \lambda)$ given by the substitution of u by $c_{NN}(z_i^r, \theta)$ in (4) and $\{z_i^r\}_{i=1}^{N_r}$ are collocation points distributed over the entire domain. The collocation points are points such that the approximation to the solution satisfies the differential equation at those points. The derivatives with respect to parameters and inputs of the network are computed using Automatic Differentiation (Baydin *et al.*, 2018).

4 The Bayesian Neural Network

The concept of Bayesian Neural Networks (BNN) was introduced in the PhD Thesis (Neal, 1995). There are some differences between the BNN and the usual Artificial Neural Networks (ANN) that deserve to be highlighted, (Goodfellow *et al.*, 2017). First of all, BNN can be used to quantify the uncertainties of the model in terms of weights and the outputs. Another important difference is related to the weights itself. In ANN, the weights belong to a single set where the aim is to optimize a loss function that best approximate the data, that is, to find optimal values of the weights. On the other hand, in the BNN, the weights are considered random variables under some distribution, and the aim is to find a marginal distribution that best fits the data, that is, the aim is to discover which distribution that best describes the weights, given the data.

Here, we consider that the input network is (z^s, λ) and the output network is $c_{NN}(z^s, \lambda, \theta)$. Since we are dealing with supervised learning, it is necessary provide target values for compare it with the network output.

Aiming provide a dataset of target values for using in the BNN framework, we have to make some planning of the samples to be taken. In this way, we consider fixed the parameters of the medium (D, V and R) and the initial and boundary conditions, where we can take N_λ samples of the vector $\lambda_i \in [0,1] \times [0,L] \times [0,T]$.

In this way, we use the Mathematica' software to generate a set of concentrations, given by the solution of (1), for each λ_j, with $j = 1, 2, \ldots, N_\lambda$, computed at the nodes $\{z_i^s\} = \{(x_i^s, t_i^s)\}_{i=1}^{N_s}$. We will name this vector by $\{c_j(z_i^s)\}_{i=1}^{N_s}$, for $j = 1, 2, \ldots, N_\lambda$.

Consider, then, the dataset of target values $\mathcal{D}$ given by

$$\mathcal{D} = \{c_j(z_i^s)\}_{N_s \times N_\lambda} \cup \{c_s(z_i^s)\}_{N_s} = \mathcal{D}_c \cup \mathcal{D}_s, \tag{6}$$

where $\mathcal{D}_c = \{c_j(z_i^s)\}_{N_s \times N_\lambda}$ is the set of target concentration values, generated by the Mathematica' software, given by the solution of (1), for each λ_j, with $j = 1, 2, \ldots, N_\lambda$, computed at the nodes $\{z_i^s\} = \{(x_i^s, t_i^s)\}_{i=1}^{N_s}$ and $\mathcal{D}_s = \{c_s(z_i^s)\}_{N_s}$ is the set of synthetically measured pollutant concentrations by sensors.

In the Bayesian Neural Network approach, we consider independent priors for the vector of parameters of the net, $P(\theta)$, and also for the parameters of the source term $P(\lambda)$, where each component of θ and λ is supposed to be a independent standard Gaussian Distribution. Observe that $P(\lambda) = \Pi_{j=1}^{N_\lambda} P(\lambda_j)$.

Therefore, given the priors $P(\lambda)$ and $P(\theta)$, we can compute the likelihood

$$P(\mathcal{D}|\, \theta, \lambda) = P(\mathcal{D}_c|\, \theta, \lambda)\, P(\mathcal{D}_s|\, \theta, \lambda),$$

where

$$P(\mathcal{D}_c|\, \theta, \lambda) = \prod_{j=1}^{N_\lambda} \prod_{i=1}^{N_s} \frac{1}{\sqrt{2\pi}\sigma^c} \operatorname{Exp}\left(-\frac{(c_j(z_i^s) - c_{NN}(z_i^s, \lambda_j, \theta)^2}{2(\sigma^c)^2}\right) \tag{7}$$

$$P(\mathcal{D}_s|\, \theta, \lambda) = \prod_{i=1}^{N_s} \frac{1}{\sqrt{2\pi}\sigma^s} \operatorname{Exp}\left(-\frac{(c_s(z_i^s) - c_{NN}(z_i^s, \lambda, \theta)^2}{2(\sigma^s)^2}\right). \tag{8}$$

The next step is to compute the posterior by using the Bayes' Theorem:

$$P(\theta, \lambda|\, \mathcal{D}) = \frac{P(\mathcal{D}|\, \theta, \lambda) P(\theta) P(\lambda)}{P(\mathcal{D})} \propto P(\mathcal{D}|\, \theta, \lambda) P(\theta) P(\lambda). \tag{9}$$

The aim here is to obtain a estimation of the parameter of the source term related to the sensors measurements, λ_s. For this purpose, we can do samples from the posterior $P(\theta, \lambda|\, \mathcal{D})$, where it can be done by using Hamiltonian Monte Carlo, (Neal, 1995), in which it is a MCMC method with a Hamiltonian Dynamics evolution that reduces the computational time for generation of the posterior random samples.

As a result, we obtain the samples $\{(\theta_i, \lambda_i)\}_{i=1}^{M}$, where by computing the mean and standard deviation of the $\{\lambda_i\}_{i=1}^{M}$, then we get a prediction for the source term λ_s, as well as the quantification of the uncertainties.

5 Numerical Experiments for PINN

In this section, we present the numerical experiments related to the PINN approach. The numerical experiments related to the BNN approach is in progress.

In the following experiments, we consider the coefficients $D = 0.06$, $V = 0.7$, $R = 0.0$ in the PDE given by Eq. (1) as well as the domain $(x, t) \in [0, 20] \times [0, 15]$. Besides, we consider fixed the intensity λ_i and the time of maximum leakage λ_t. The PINN was implemented using the open-source framework TensorFlow (Abadi *et al.*, 2015) with the Python application software interface (API). However, in view of limited CPU, we choose to use the Google Colab, in which is a way of write and execute Python codes directly in the internet browser, even without any Python software installed and with access to GPUs free of charge in a free account[1].

[1] https://colab.research.google.com/

5 1 Estimation of source location

In this subsection, we consider $\lambda_i = 1.7$ and $\lambda_t = 5.3$ and implement the source location estimation with the PINN approach.

In the Tab. 1, the numerical experiments related to the topology of the network are presented without the action of noise in the measurements of the sensors. Besides, we consider as activation function the Leaky RELu, with slope $\alpha = 0.1$, the number of samples in the x-direction and t-direction are fixed in 50 each, that is, $N_s = 50 \times 50 = 2500$ and the number of collocation points fixed in 1000, that is, $N_r = 1000$. The number of epochs is fixed in 250 and the initial guess of $\lambda_l = 7.0$. In all experiments, we use ADAM optimizer with learning rate varying accordingly of epoch of the following way $\delta(n) = 0.01(1 - H(n-100)) + 0.001(H(n-100) - H(n-200)) + 0.0001H(n-200)$.

We implement several experiments varying the number of layers and the number of neurons in each layer, where the best results are presented in the Tab. 1. Observe that the all the predictions presented were close to the real location value, with relative error lower than 0.2%. The best prediction was the one with 3 layers and 40 neurons. In this way, in the next experiments, we will consider only networks of 3 layers with 40 neurons.

Table 1. Estimation of Source Location - PINN - 0% Noise - $N_s = 2500$ measured points, $N_r = 1000$ collocation points

Layers	Neurons	Exact Location (λ_l)	Predicted Location (λ_s)	Standard Deviation	Rel. Error ($\%$)
3	40	7.8	7.7942	0.07749	0.074
4	30	7.8	7.8066	0.04718	0.085
6	40	7.8	7.7883	0.14246	0.150

In the Tab. 2, we only consider the net with 3 layers with 40 neurons and noisy measurements. We keep the same measured points and collocation points as in the previous experiment. As expected, higher the noise level in the measurements of the sensors, greater the relative error in the prediction.

Table 2. Estimation of Source Location - PINN - Noisy Measurements - $N_s = 2500$ measured points, $N_r = 1000$ collocation points

Noise ($\%$)	Exact Location (λ_l)	Predicted Location (λ_s)	Standard Deviation	Rel. Error ($\%$)
0	7.8	7.7942	0.07749	0.074
1	7.8	7.8184	0.10714	0.236
2	7.8	7.7613	0.07251	0.496
5	7.8	7.8604	0.06566	0.774
10	7.8	7.5404	0.68362	3.328

6 Partial Conclusions

In this work in progress, we study the inverse problem of estimation of pollution sources by using the Physics-Informed Neural Network and the Bayesian Neural Network.

First, we consider to reconstruct only the location of the source term by PINN. A next step is also consider the other source parameters as the intensity of the source and the time of maximum leakage.

The direct problem was solved by classical numerical methods with the goal of generating a dataset to be used in the neural network approach for the inverse problem.

The inverse problem was solved by a special neural network, namely Physics-Informed Neural Network (PINN). The PINN was capable to recover the location of the source term from sensors measurements, even in presence of noise.

The next step is to implement more experiments related with the PINN, varying the number of measured and collocation points, and also to study some strategy of planning the distribution of the points $z^s = (x^s, t^s)$ and the values of λ_i. Some works suggest that each sum could have a weight that gives more or less importance in the optimization process (van der Meer, 2019), (Rojas *et al.*, 2021). Besides, the Bayesian Neural Network will be implemented in order to compare the results and the computational time necessary to provide the parameters prediction.

Another way of extending this work is to consider the Bayesian Neural Network combined with the Physics-Informed Neural Network (Yang *et al.*, 2021), in which the physical laws can serve as an additional prior and the Hamiltonian Monte Carlo can continue serve as an estimator of the posterior.

7 ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, National Council for Scientific and Technological Development - CNPq and FAPERJ - Carlos Chagas Filho Foundation for the Research Support of the State of Rio de Janeiro. H.S.M. also acknowledge the Rio de Janeiro State University (UERJ), for the PAPD Program and FAPERJ - Carlos Chagas Filho Foundation for the Research Support of the State of Rio de Janeiro for the Emeritus Visiting Researcher grant.

8 REFERENCES

Abadi, M., Agarwal, A., Barham, P. and et. al, 2015. "Tensorflow: Large-scale machine learning on heterogeneous systems". Software available from tensorflow.org.

Andrle, M. and Badia, A.E., 2015. "On an inverse source problem for the heat equation. application to a pollution detection problem, ii". *Inverse Problems in Science and Engineering*, Vol. 23, No. 3, pp. 389–412.

Baydin, A.G., Pearlmutter, B.A., Radul, A.A. and Siskind, J.M., 2018. "Automatic differentiation in machine learning: a survey". *Journal of Machine Learning Research*, Vol. 18, pp. 1–43.

Colton, D. and Kress, R., 2013. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer, New York, 3rd edition.

El Badia, A., Ha-Duong, T. and Hamdi, A., 2005. "Identification of a point source in a linear advection–dispersion–reaction equation: application to a pollution source problem". *Inverse Problems*, Vol. 21, pp. 1121–1136.

Evans, L.C., 1998. *Partial Differential Equations*. Americal Mathematical Society, 1st edition.

Goodfellow, I., Bengio, Y. and Courville, A., 2017. *Deep Learning*. MIT Press, Cambridge, MA, 1st edition.

Hadamard, J., 1902. "Sur les problèmes aux dérivées partielles et leur signification physique". *Princeton University Bulletin*, pp. 49–52.

Isakov, V., 1990. *Inverse Source Problem*. American Mathematical Society, Providence, Rhode Island, 1st edition.

Kirsch, A., 2011. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer, New York, 2nd edition.

Moura Neto, F.D. and Silva Neto, A.J., 2013. *An Introduction to Inverse Problems with Applications*. Springer, New York, 1st edition.

Neal, R.M., 1995. *Bayesian Learning for Neural Networks*. Ph.D. thesis, University of Toronto, Toronto, Canada.

Raissi, M., Perdikaris, P. and Karniadakis, G.E., 2019. "Physics-informed neural networks: A deep learning framework for solving forward and inverse probems involving nonlinear partial differential equations". *Journal of Computational Physics*, Vol. 378, pp. 686–707.

Rojas, C.J., Bittencourt, M.L. and Boldrini, J.L., 2021. "Parameter identification for a damage model using a physics informed neural network". *arXiv:2107.08781v1*, pp. 1–31.

Tarantola, A., 2005. *Inverse problem theory and methods for model parameter estimation*. SIAM, Philadelphia, PA, 1st edition.

van der Meer, R., 2019. *Solving partial diferential equations with neural networks*. Master's thesis, Delft University of Technology, Delft, Netherlands.

Yang, L., Meng, X. and Karniadakis, G.E., 2021. "*B - PINN*: Bayesian physics-informed neural networkss for forward and inverse pde problems with noisy data". *Journal of Computational Physics*, Vol. 425, No. 109913.

9 RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.