encit 2022

ABCM Associação Brasileira de Engenharia e Ciências Mecânicas

19th Brazilian Congress of Thermal Sciences and Engineering
November 6th–10th, 2022, Bento Gonçalves - RS - Brazil

ENC-2022-0108
ESTIMATION OF POLLUTION SOURCES WITH PHYSICS-INFORMED 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

Vinicius Albani
Universidade Federal de Santa Catarina. R. Eng. Agronômico Andrei Cristian Ferreira, s/n - Trindade, Florianópolis - SC, 88040-900
v.albani@ufsc.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.
In this work, the inverse problem of identification of pollution sources location and intensity in a river is studied considering the advection-dispersion equation, along with a Neural Network approach. In the direct problem, the location and the intensity of the source terms are known as well as the PDE coefficients and then it is solved by classical numerical methods. The numerical solution is computed in random nodes in the domain with the aim of to generate a dataset representing a synthetic experimental data, where it will be used as input, noisy included, for the inverse problem and formulation. The inverse problem is posed as, given some measurements from sensors (synthetically generated) and the medium parameters, we want to estimate the pollutant source location. This problem is solved by two networks: the usual Artificial Neural Network (ANN) and the Physics-Informed Neural Network (PINN). In both cases we consider a network formed by Multi-Layer Perceptron in a fully-connected network. In the ANN approach, we construct a net that associates the concentration at the nodes as input to the source location as a output. First, we train the net, obtaining a optimal set of weights, and in a second step, we apply the net at the sensor measurement. This calculation is instantaneous and generate an estimation of source location. The numerical experiments using ANN are implemented, showing good results, even when considering noisy measurements. In the other hand, in the PINN approach, the measurements from the sensors are incorporated in the new loss function, where in this work, we also consider initial and boundary conditions in this optimization process. The PINN is a recent type of neural networks that are trained to solve some learning tasks while, at the same time, satisfies the physical laws that describes the phenomena involved. The present approach of constructing the loss function with points beyond those from measurements is slightly different from the original research paper that introduces the PINN framework. Numerical experiments related to estimation of source location are presented, with results not so good as those from ANN. Further experiments are necessary to improve the results of the present work.

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

1. INTRODUCTION

Over the last centuries, with the development of industrial production, started with the industrial revolution, the world was capable to produce in larger quantities, to improve the quality process and to reduce the time of goods production. However, with this increasing industrial process, the pollution had also increased, generating, in many cases, contaminants in the air, soil or water. Unfortunately, the leakage of pollutant in the water like rivers, lakes or ocean is a common fact nowadays. This kind of pollution source includes the industrial rejects, the oil refinery 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 a allied 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 to determine 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, (Neto and Neto, 2013). In the example above, the direct problem can be formulated as, given the parameters of the medium and the location of the source pollution term, we are interested in measure the pollution concentration on the riverside region.

The direct problem is well-posed problem in 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 type considered in this work is the Physics-Informed Neural Network (PINN).

The PINN is a recent type of neural networks introduced in (Raissi *et al.*, 2019). These nets has the characteristic that it takes into account the physical laws that describes the phenomena involved while, at the same time, train to solve some learning tasks. It can be made by modifying the Mean Squared Error Loss (the function where we are interested in minimize 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 PDE's, 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).

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 generate dataset to be used in the inverse problem framework with neural networks considering the equation (PINN).

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 problem considered for the pollutant source concentration. The Section **??** is devoted to formulation of the PINN for the inverse problem. In Section 5.numerical experiments are presented related with the identification of the source location and the parameter estimation of the reaction term. 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.

2.1 General Mathematical Formulation

Let Ω be an open, bounded subset of $\mathbb{R}^2$ (it could be adapted for $\mathbb{R}^N$), with regular boundary $\partial\Omega$, representing a river of length L and width l. In the context of dispersion of pollutants in a river, we define the function $c(x,t)$ as the pollutant concentration, for $x \in \Omega$ and $t > 0$, posed by the two-dimensional advection-dispersion-reaction equation

$$\begin{cases} c_t - D\Delta c + V \cdot \nabla c + Rc = F(x,t), \text{ for } \; x \in \Omega, 0 < t < T \\ c(x,0) = g_0(x), \text{ in } \Omega \\ c(x,t) = g(x,t), \text{ on } \partial\Omega \times [0,T] \end{cases} \tag{1}$$

where the subscript t in c_t represents the partial derivative of c with respect to time, Δc represents the Laplacian of function c and ∇c represents its gradient vector. Beside this, D is the diffusion coefficient, V is the velocity vector, R is the reaction coefficient, F is the source term, g_0 is the initial concentration and g is the known concentration values at the boundary. This problem has a unique solution, provided that u, F, g_0 and g belong to appropriated function spaces, see for example (Evans, 1998).

The Direct Problem for (1) has the aim of, given the coefficients of the PDE, the boundary and initial 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, a Inverse Problem for this general setting is, given the boundary measurements g, 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 difficulty, we must assume that the source term F, that we expect to reconstruct, belongs to a special class of function.

In the following, it will be established the specifics direct and inverse problems that will be considered in the rest of the work.

2.2 The Uni-dimensional Problem Considered

The two-dimensional problem considered in (1) takes into account the length and the width of the river under analysis. However, if the river current is supposed to be in the x-direction and that the pollutant concentration tends to flow more in x-direction, then it is possible to reduce the original problem of two spatial dimensions to one spatial dimension.

In this way, we consider the following one-dimensional problem

$$\begin{cases} c_t - Dc_{xx} + Vc_x + Rc = F(x,t), \text{ for } \ x \in (0,L), t \in (0,T) \\ c(x,0) = g_0(x), \text{ for } x \in [0,L] \\ c(0,t) = g_l, \text{ for } t \in [0,T] \\ c(l,t) = g_r, \text{ for } t \in [0,T]. \end{cases} \tag{2}$$

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 concentrations that change along the time.

Beside this, we also consider the source term $F(x,t)$ in Eq. (2) given by

$$F(x,t) = \lambda_1 \left(e^{-(x-\lambda_2)^2} \right), \text{ with } \ 0 < a < L, \tag{3}$$

where $\lambda_1 > 0$ is the intensity of the source pollutant concentration and $\lambda_2 \in [0,L]$ is the source location. This kind of source represents a continuous punctual leakage scenario concentrated in position λ_2 and with decreasing intensity when we analyse a position far away of λ_2.

2.3 The Direct and Inverse Frameworks Overview

The Direct Problem for (2) is posed as: given the vector of parameters $\lambda = (\lambda_1, \lambda_2)$, we must to find the solution $c_\lambda(x,t)$. Here we consider that the intensity is the same for all possible locations. This problem is, then, solved with the Finite Element Method through built-in functions in the *Mathematica* software, generating a synthetic dataset.

The Inverse Pollution Source Problem studied here is posed as: given the PDE parameters in Eq. (2), the initial/boundary conditions g_0, g_l and g_r, and supposing that the source term F that we expect to reconstruct is given by Eq. (3), with $\lambda_1 = 1$, the aim is to discover the source location λ_2, from the concentration measurements.

These frameworks will be detailed in the following.

3. The Artificial Neural Network Approach

In this section, we present the direct problem and the dataset generated, the usual Artificial Neural Network (ANN), formed by a Multi-Layer Perceptrons (MLP), and the inverse problem considered for this neural net.

3.1 Direct Problem and the Dataset Generated

Consider the Direct Problem (2). In order to reduce the problem dimension, let us consider the intensity as unitary, that is, $\lambda_1 = 1$ and, beside this, the location λ_2 as number between 0 and L. Without loss of generality, we write $\lambda = \lambda_2$.

For different source location λ_j, $j = 1, 2, \ldots, N_\lambda$, where N_λ is the number of source locations to be considered in the training process, we compute $F_j(x,t)$, from (3). So, the direct problem is solved with the Finite Element Method built-in functions in *Mathematica*' software, generating a numerical solution $c_j(x,t)$. Figure 1 shows an example of this numerical solution for pollution source with intensity $\lambda_1 = 1$, centred at location $\lambda_2 = 3.7$m.

In this way, if we consider $N_x + 1$ points in the interval $[0,L]$, with $\Delta x = L/N_x$, and $N_T + 1$ point in the time interval $[0,T]$, with $\Delta t = T/N_T$, we can define the nodes

$$x_k = x_0 + k\Delta x, \text{ with } k = 1, 2, \ldots, N_x, \ x_0 = 0 \ \text{ and } \ x_{N_x} = l,$$

$$t_n = t_0 + n\Delta t, \text{ with } n = 1, 2, \ldots, N_T, \ t_0 = 0 \ \text{ and } \ t_{N_T} = T.$$

So, for each λ_j, $j = 1, 2, ..., N_\lambda$, where we supposed sampled from an Uniform Distribution over the interval $[0,L]$, we can determine the numerical solution evaluated in each node (x_k, t_n), called $(c_j(x_k, t_n))$. Observe that this tensor dataset has dimension $N_\lambda \times N_x \times N_T$.

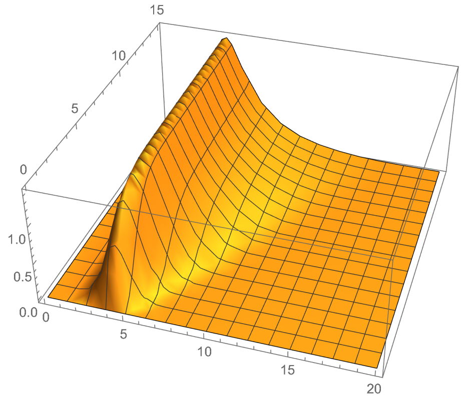

Figure 1. Numerical Solution of the direct problem (2), with F given by Eq. (3), considering D = 0.06, V = 0.7, R = 0.3, $g_0 = g_l = g_r = 0$.

In this way, the full dataset is given by

$$\{c_j(x_k, t_n), \lambda_j\}_{j=1}^{N_\lambda}, \tag{4}$$

where $k = 1, 2, ..., N_x$ and $n = 1, 2, ..., N_T$.

3.2 The Artificial Neural Network - ANN

Consider a fully-connected neural network with $Q \geq 1$ hidden layers as the surrogate model. The $j-th$ hidden layer, $z_j \in \mathbb{R}^{N_m}$, for $j = 1, 2, ..., N_Q$, and the output of the neural network, $\tilde{\lambda} \in \mathbb{R}$, are given by

$$z_j = \phi\left(w_{j-1}.z_{j-1}\right) + b_{j-1}, \; j = 1, 2, ..., Q, \tag{5}$$

$$\tilde{\lambda} = w_Q.z_Q + b_Q, \tag{6}$$

where w_j are the weight matrices and b_j the biases vectors for the hidden layer j, $\phi(\cdot)$ is the nonlinear activation function, where we use the hyperbolic tangent function and the Leaky Rectified Linear Unit (Leaky ReLU) function (Maas *et al.*, 2013). Beside this, the input data is given by $z_0 = \{c(x_k, t_n)\}_{N_x.N_T}$, $N_0 = N_x.N_T$ and $N_{Q+1} = 1$.

Observe that, in the full dataset defined previously in Eq. (4), we can split it in another two sets:

$$\{c_j(x_k, t_n), \lambda_j\}_{j=1}^{N_\lambda} = \{c_j(x_k, t_n), \lambda_j\}_{j=1}^{Ntrain} \bigcup \{c_j(x_k, t_n), \lambda_j\}_{j=1}^{Ntest},$$

where we named a training set and a test (validation) set. The training set considered is 70% of the full dataset and the remaining 30%, the test set, that is, $Ntrain = \lfloor (0.7).N_a \rfloor$ and $Ntest = \lfloor (0.3).N_a \rfloor$, where $\lfloor x \rfloor$ is the least integer function that gives the greatest integer that is less than or equal to x.

The training set is defined as a vector of association

$$\{c_j(x_k, t_n)\}_{N_x.N_T} \rightarrow \lambda_j,$$

for $j = 1, 2, ..., N_a$. This set will be used to inform the neural net that, when we have a pollutant concentration $c_j(x_k, t_n)$, it is generated by a source of type (3), with intensity constant equal to 1, located at the point λ_j. For simplicity in the code for MLP, used in this work, we write the input as a vector of length $N_x \cdot N_T$. In the case of consider the matrix form, it is possible to use the net of convolutional type.

The test set is defined in a similar way with the remaining dataset. This set is important to measure or "to test" if the weights and biases find in the optimization process from the training set is good enough to predict a value that is comparable to those in the test set. In another words, after the training process, we can compare

$$|\lambda_j^{predict} - \lambda_j^{exact}|,$$

where $\lambda_j^{predict}$ is the computed output of net, Eq. (6), when the input is $c_j(x_k, t_n)$ (from the test set) and λ_j^{exact} is the expected correct value (from the test set). If this difference is not small enough, then the net continue the optimization procedure. After this process, we find the weight matrix, w, and the biases vector, b. Consequently, we find the function $\tilde{\lambda}$, given by Eq. (6).

The details about the topology of neural network considered in this work as well as the activation and the loss functions will be given in the numerical experiments section.

3.3 The ANN for Inverse Problem

As commented before, the goal in this work is to estimate the location of a pollutant source in a river of length L, from certain measurements extracted from sensors.

Here, we consider N_{xs} sensors distributed randomly through a Uniform Distribution over $[0,\ L]$, where each one is capable of collect information as many times as we want. In this way, a single sensor can provide, for example, N_{ts} samples of pollutant concentration, that is, we have a total of $(N_{xs}) \times (N_{ts})$ samples of concentration.

Since the input dimension of the neural net is $N_x.N_t$, we must consider that the number of sensor, N_{xs}, and the number of samples collected for each one, N_{ts}, must to satisfy the relation

$$(N_{xs}) \times (N_{ts}) = (N_x) \times (N_t).$$

In this way, the procedure to solve the inverse source location problem is described below.

Firstly, we define the dataset for training and testing the neural network and realize the training process. After this, we obtain the weights and biases that optimize the loss function, where we considered the Mean Squared type.

Second, we collect the measurements concentrations from the N_{xs} sensors. This collection can be from a experimental or a synthetic point of view. In this work, we generate synthetic data from the Mathematica' Solver, that is, we solve the direct problem to obtain the numerical solution. This solution is, then, computed in the nodes (x_s^k, t_s^n), for $k = 1, 2, \ldots, N_{xs}$ and $n = 1, 2, \ldots, N_{ts}$, where x_s^k is the k-th sensor location in which measures the pollutant concentration at the times t_s^n. The sensor location can be taken randomly.

Third, we compute the output $\lambda^{predict}$, considering as input of the net the measurements of the sensors already calculated in the second step. Since we are dealing with random quantities, we realize this step five times, in view to obtain a more reliable results.

Forth, we take the mean and standard deviation of the previous results locations, compare the mean with the exact value and compute the relative error

$$\frac{|\lambda^{predict} - \lambda^{exact}|}{\lambda^{exact}}.$$

4. The Physics-Informed Neural Network Approach

As mentioned in Introduction, the Physics-Informed Neural Network (PINN) is a recent type of neural network introduced recently that has the characteristic that, in the optimization process, we search for the weights and biases that minimizes a special modified loss function.

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

4.1 The PINN for Inverse Problem

Here we will point some differences when construct an architecture for a deep neural network. When we are dealing with ANN architecture, we have certain "freedom" to define what are the inputs and the outputs of the net. Depending on the architecture, we can modified this ANN and study a Convolutional Neural Network or a Recurrent Neural Network. For example, in the previous sections, we define that the pollutant concentration at some nodes would be considered as input and the source location as the output. In the PINN approach, this architecture will change slightly.

Suppose that we have fixed a source location (unknown) λ in the inverse problem context. We could consider another ANN whose inputs are the nodes (x_k, t_n), for $k = 1, 2, \ldots, N_x$ and $n = 1, 2, \ldots, N_t$, and the outputs are the concentration $c(x_k, t_n)$. The Mean Squared Error (MSE) Loss Function for this net would be given by

$$MSE = \frac{1}{N}\sum_{i=1}^{N} |\tilde{c}(x_i, t_i) - c_i^e|^2, \tag{7}$$

where $\tilde{c}(x_i, t_i)$ is the neural network computed in the node (x_i, t_i) and c_i^e is the exact value that can be from the training/test set.

On the other hand, if we define the function

$$f = c_t - Dc_{xx} + Vc_x + Rc - F_\lambda(x, t), \tag{8}$$

where $F_\lambda(x, t)$ is given by (3) with λ unknown, and proceed by approximating the pollutant concentration $c(x, t)$ by a deep neural network $\tilde{c}$, then this will result in a different neural network for $f(x, t)$, namely Physics-Informed Neural Network (Raissi *et al.*, 2019).

It is important to point out that we suppose that the neural networks $\tilde{c}$ and f have parameters shared between them.

So, this shared parameters and the unknown λ can be optimized (or learned) by minimizing the new loss function

$$MSE = \frac{1}{N_p}\sum_{i=1}^{N_p}|\tilde{c}(x_i^p,t_i^p)-c_i^e|^2+\frac{1}{N_c}\sum_{i=1}^{N_c}|f(x_i^c,t_i^c)|^2, \tag{9}$$

where

$$f(x_i^c,t_i^c)=\tilde{c}_t(x_i^c,t_i^c)-D\tilde{c}_{xx}(x_i^c,t_i^c)+V\tilde{c}_x(x_i^c,t_i^c)+R\tilde{c}(x_i^c,t_i^c)-F_\lambda(x_i^c,t_i^c), \tag{10}$$

F_λ is given by Eq. (3), $\{x_i^p,t_i^p\}_{i=1}^{N_p}$ are training points, that is, the points that correspond the initial and boundary conditions as well as the measurements from the sensors, that is,

$$\{x_i^p,t_i^p\}_{i=1}^{N_p}=\{x_i^0,0\}_{i=1}^{N_0}\bigcup\{x_i^b,t_i^b\}_{i=1}^{N_b}\bigcup\{x_i^s,t_i^s\}_{i=1}^{N_s}, \tag{11}$$

with N_0, the number initial points, N_b, the number of boundary points and N_s, the number of measurements in entire domain. Observe that the above set splitting generates different loss summation in Eq. (9), L_0, L_b, L_s and L_r, with loss relative to the initial, boundary, sensors and collocation points, respectively. This approach for the considered data is slightly different from those presented in (Raissi *et al.*, 2019), in which only considers the measurement data. Figure 2 represents an example of training set, where every positions were sampled from a Uniform Distribution over each domain considered.

Figure 3 represent the flowchart of the methodology adopted in this work, where we create a neural network with (x,t) as inputs and the concentration $\tilde{c}$ as output. This output is derived using Automatic Differentiation(Baydin *et al.*, 2018), in which we can compute the residue (10). After this, we computed four losses, each one considering different points of the domain (11), where the sum of the losses is minimized for the parameters λ (PDE parameter) and (w,b) (neural net parameters).

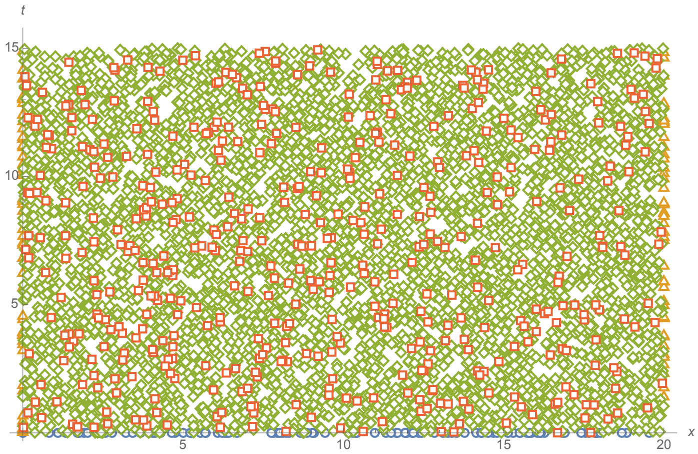

Figure 2. Training Data with $L = 20$ and $T = 15$. 100 initial points (Blue Circle), 100 boundary points (Orange Triangle), 500 measurement points (Red Square) and 10000 collocation points (Green Diamond)

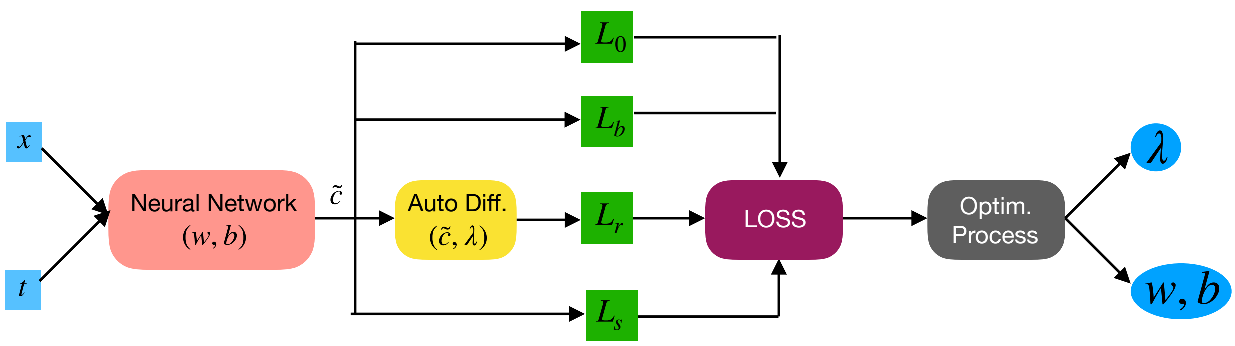

Figure 3. Flowchart for the Parameter Estimation using the PINN approach.

5. Numerical Experiments

In this section, we present some numerical experiments related to the estimation of a location of pollutant source in a river, posed by the advection-dispersion equation, that is, we consider null the reaction coefficient.

5.1 Parameter Estimation in the ANN

In this subsection, we implement the source location estimation from the ANN. In the following experiments, we consider the coefficients $D = 0.06$, $V = 0.7$, $R = 0.0$ in the PDE given by Eq. (2) as well as the domain $(x, t) \in [0, 20] \times [0, 15]$. Beside this, we consider that there are not initial pollutant concentration in the river, neither pollutant concentration in the boundary.

The experiments for ANN were implemented in the Mathematica, version 12.2.0.0, in a macOS 10.13.6, with 8GB RAM, without a GPU. In the Tables 1, 2 and 3, we do not write the number of neurons in each layer due the limitations of the Mathematica.

In this experiment, we consider the exact source location $\lambda_2 = 3.7$. For generate a training/test set, we consider $N_\lambda = 40$, that is, we consider 40 possible source location, λ_j, in the space interval [0,20]. Beside this, we consider 70% of the full dataset as a training set and the remaining 30%, the test set. For each λ_j, $j = 1, 2, \ldots, N_\lambda$, we solve the problem (2) and computed this numerical solution in the nodes (x_k, t_n), $k = 1, 2, \ldots, N_x$ and $n = 1, 2, \ldots, N_t$, generated randomly from a Uniform Distribution over the entire domain. So, the problem (2) is solved via Finite Element Method built-in function in Mathematica, Following the ideas from Section 3, we create an ANN with input of $(N_x) \times (N_t)$ and output of dimension 1, representing the training set (4).

In the Tab. 1, we present the results varying the number of hidden layers of hyperbolic tangent activation function type, but considering no noise, the number of samples in the x-direction , t-direction and the number of measurements also fixed in 100. In all experiments, we use ADAM optimizer, batch size 28, 10000 epochs. We can note that the relative error when we consider 4 layers is lower than when we consider 6 layers, even run multiple times. This fact can be as result of some overfiting issue.

Table 1. Estimation of Pollutant Source Location - Artificial Neural Network - 0% Noise

Layers	Exact Location	Predicted Location	Rel. Error
2	3.7	3.67838	0.58%
3	3.7	3.68089	0.52%
4	3.7	3.69888	0.03%
6	3.7	3.68945	0.29%

In the Tab. 2, we present the results also varying the number of hidden layers of hyperbolic tangent activation function type, but with 0.5% of noise in the sensors. We repeat all experiments 30 times and take the mean of the location and of the relative error.

Table 2. Estimation of Pollutant Source Location - Artificial Neural Network - 0.5% Noise

Layers	Exact Location	Predicted Location	Rel. Error
2	3.7	3.66619	0.91%
3	3.7	3.66876	0.84%
4	3.7	3.71125	0.51%
6	3.7	3.69484	0.49%

In the Tab. 3, we present the results also varying the number of hidden layers of hyperbolic tangent activation function type, but with 1% of noise in the sensors. Again we repeat all experiments 30 times and take the mean and standard deviation of the relative error.

Table 3. Estimation of Pollutant Source Location - Artificial Neural Network - 1% Noise

Layers	Exact Location	Predicted Location	Rel. Error
2	3.7	3.66533	1.29%
3	3.7	3.66804	1.14%
4	3.7	3.6797	1.02%
6	3.7	3.67898	1.07%

5.2 Parameter Estimation in the PINN

In this subsection, we implement the source location estimation from the PINN. Again, we consider the same PDE coefficients, domain, initial and boundary condition and exact source location as in the ANN case.

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 use the Google Colab, that 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].

An important point that distinguish the two approach is the measurements from sensors. In the ANN example, we consider a few values from sensors, different from this example, that we will consider some higher number of measurements. Another difference rely on the fact that in the PINN we incorporate the measurements from sensors in the training process.

In the Tables 4 and 5, we consider only network with 4 layers in view of in the ANN approach this generated better results, but we test different numbers of neurons to verify the accuracy.

In the Tab. 4, we consider as activation functions the Leaky RELu, with slope $\alpha = 0.1$, but considering no noise, the number of samples in the x-direction and t-direction are fixed in 100, the number of collocation points fixed in 2000 and the number of measurements also fixed in 500. In all experiments, we use ADAM optimizer with learning rate varying accordingly of epoch of the following way $\delta(n) = 0.01(1 - H(n - 1000)) + 0.001(H(n - 1000) - H(n - 3000)) + 0.0001H(n - 3000)$. The number of epoch is fixed in 3000 and the initial guess of $\lambda_2 = 5.0$.

Table 4. Estimation of Pollutant Source Location - Physics-Informed Neural Network - 0% Noise - 500 measured points

Layers	Neurons	Exact Location	Predicted Location	Standard Deviation	Rel. Error
4	20	3.7	3.5994174	3.0194862	2.72%
4	40	3.7	3.8405252	2.9921703	3.80%
4	60	3.7	4.067301	2.3033612	9.93%

In the Tab. 5, we keep all the features of the previous table and the number of measurements fixed in 2000, but we.

Table 5. Estimation of Pollutant Source Location - Physics-Informed Neural Network - 0% Noise - 2000 measured points

Layers	Neurons	Exact Location	Predicted Location	Standard Deviation	Rel. Error
4	20	3.7	2.0105023	3.8876734	45.66%
4	40	3.7	2.9883664	3.386964	19.23%
4	60	3.7	4.886573	3.7313771	32.07%

We note that PINN generated results that were in principle worse than the ones from the ANN. We intend continue this study to understand what could be causing this effect, as an example of possible issues we can cite a too many parameters to optimize or we have to ponder the losses (9) or even a overfitting problem.

6. Conclusions & Extensions

In this work, we study the inverse problem of estimation of location of pollution sources by using an usual Artificial Neural Network and the Physics-Informed Neural Network.

First, we consider to reconstruct only the location of the source term by ANN and PINN. The next step is also consider the intensity of the source, where we will consider the case where $\lambda_1 \neq 1$.

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, that is, this dataset was split in two sets, a training set and a test set, where the first set will be used to train the network for the inverse problem.

The inverse problem was solved by the two kinds of neural network. The firs type considered, the ANN, was capable to recover the location of the source term from sensors measurements, even in presence of noise. On the other hand, the PINN proved to be more sensible to the number of sensors measurements as well as the architecture of the net.

The next step is to improve the PINN estimation of the source location and statistical studies will be implemented to bring more accuracy to the work. This kind of neural net is known to be more accurate than the usual ANN, even with few data. Other possible issue is related to the order of importance that each loss term represent in the total loss function. Some works suggest that each sum could have a weight that gives more or less importance in the optimization process (van der Meer, 2019).

Another way of extending this work is to consider the Bayesian Neural Network combining with the Physics-Informed Neural Network (Yang *et al.*, 2021), in which we can adopt a different sampling method of posterior (Rojas *et al.*, 2021).

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

7. ACKNOWLEDGEMENTS

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

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.

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.

Maas, A.L., Hannun, A.Y. and Ng, A.Y., 2013. "Rectifier nonlinearities improve neural network acoustic models". In *Proceedings of the 30 th International Conference on Machine Learning*. Atlanta, Georgia, USA.

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

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.