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-2023-1625

Optimizing Contaminant Source Identification with MLP Neural Network

Guido F. M. G. de Carvalho
Antônio J. da Silva Neto
Diego C. Knupp
Universidade do Estado do Rio de Janeiro, Brazil
{guido.fraga, ajsneto, diegoknupp}@iprj.uerj.br

David A. Pelta
Universidad de Granada, Spain
dpelta@ugr.es

Abstract. *Identifying the source of contaminants in the atmosphere is of great importance across diverse fields such as agriculture, industry, ecology, and security. This study proposes a faster solution to the source identification inverse problem by leveraging the use of a Multi-Layer Perceptron (MLP) neural network as an efficient solver of the contaminant dispersion problem. The MLP network was trained using data from a numerical solution to the two-dimensional advection-diffusion equation, considering different contaminant source locations. This approach presented a significantly reduced computational time, when compared to the conventional numerical methods. The results demonstrated promising performance with an average distance of approximately* 10^{-2} *for 40 estimated cases. This method has the potential to improve the efficiency of contaminant source identification systems, making it valuable for various applications, including real-time monitoring systems.*

Keywords: *Inverse Problem, Source Identification, Neural Networks, Advection-Diffusion Equation, Optimization*

1. INTRODUCTION

Pollution, including air, water, and soil contamination, poses significant threats to ecosystems, biodiversity, and human well-being. Identifying the sources of pollutant distribution is crucial for devising effective strategies to control and mitigate pollution. Such approaches enables the implementation of targeted measures to minimize their impact (Luo *et al.*, 2019). Accurately identifying pollutant sources, similar to detecting furtive gas leaks, holds significant security implications. In both cases, pinpointing the source of the substance enables the implementation of effective mitigation measures to prevent further release and potential harm. By swiftly identifying the source, targeted actions can be taken to reduce or eliminate the release of pollutants or hazardous gases. This proactive approach not only ensures the security of communities but also mitigates potential risks and protects against adverse consequences.

A common method for simulating the spread of substances in the atmosphere involves employing the advection diffusion equation. This equation characterizes the movement of a substance through the flow of a fluid while considering the simultaneous diffusion of the substance within the fluid medium, such as air (Stockie, 2011). Solving this equation presents challenges due to the computational-intensive nature of solving numerically its partial differential equation. Traditionally, unconditionally stable techniques, such as the implicit approach of the Finite Difference Method (FDM), have been utilized to address this problem (Silva Neto and Becceneri, 2009; Liu *et al.*, 2007). Although this approach is numerically stable, it usually leads to significant computational costs, resulting in longer solving times. Moreover, as the mesh complexity or its refinement escalates, the computational cost of solving the equation system can become prohibitively expensive. Hence, more efficient algorithms are necessary to achieve more efficient solutions for such problems.

The source identification problem can be defined as an inverse problem, where the objective is to identify the parameters that resulted in the observed data. Inverse problem solution methods typically involve solving the direct problem multiple times, so the optimal parameters that match the provided data are identified. However, if the direct problem solution is computationally demanding, it significantly increases the time spent solving the inverse problem. In some cases, this can render the inverse problem solution infeasible or too slow, particularly in time-critical scenarios like emergency situations. One approach to address this challenge is to utilize a Multilayer Perceptron (MLP) network as the direct problem solver. By employing an MLP network, the direct problem solution becomes more efficient, leading to faster and more feasible solutions for the inverse problem. This enhancement enables quicker response times and more effective decision-making in critical situations.

The paper is organized as follows: Section 2 presents the direct problem, which is divided into two subsections: Physical Problem and Artificial Neural Network Model, describing the mathematical model and numerical method used

to generate synthetic data, and the ANN trained with the generated data to solve the direct problem. Section 3 presents the inverse problem algorithm used. Section 4 shows the results obtained by the combination of methods presented in Sections 2 and 3, for various different cases simulated. Finally, the Conclusion section summarizes the main contributions of the paper and proposes future research directions.

2. DIRECT PROBLEM

The direct problem discussed in this work focuses on the bidimensional advection diffusion model. This section offers an overview of this problem, emphasizing the understanding of its physical aspects and the utilization of neural network techniques for more efficient prediction of the substance's behavior.

2.1 Physical Problem

The bidimensional advection diffusion problem presents a mathematical model that describes the dynamics of a substance transported by a fluid in a two-dimensional setting. This problem is encountered in various physical systems, including the dispersion of pollutants in the atmosphere or the diffusion of chemicals in porous media. Two fundamental mechanisms are involved in this model: advection, representing the substance's transportation by the fluid, and diffusion, signifying the dispersion of the substance due to its concentration gradient (Moreira *et al.*, 2009; Prieto *et al.*, 2011).

To mathematically describe this process, we employ the two-dimensional advection-diffusion equation. This equation, formulated in terms of the substance concentration $c(x, y, t)$, can be written as

$$\frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c = D \nabla^2 c, \tag{1}$$

where $\mathbf{v}$ denotes the velocity vector of the fluid, D represents the diffusion coefficient, and ∇ signifies the gradient operator. By utilizing this equation, we can comprehend the alterations in the substance's concentration over time, owing to the combined influence of advection and diffusion effects. The solution of this equation results in the concentration distribution in space and time within a given domain.

In this study, second-type contour conditions were used as boundary conditions. These conditions establish that the derivative of the solution with respect to the normal direction is set to zero at the domain boundaries $[0, L_x] \times [0, L_y]$. This assumption implies that there is no substance flux across the boundary, which is a plausible assumption for many practical problems.

These conditions can be expressed as:

$$\frac{\partial c}{\partial x}(0, y, t) = \frac{\partial c}{\partial x}(L_x, y, t) = 0, \quad \text{for } 0 \leq y \leq L_y,\ t \geq 0, \tag{2}$$

$$\frac{\partial c}{\partial y}(x, 0, t) = \frac{\partial c}{\partial y}(x, L_y, t) = 0, \quad \text{for } 0 \leq x \leq L_x,\ t \geq 0. \tag{3}$$

To model the initial concentration of a substance released from a point source, a bell curve approach was employed. The Gaussian formula is written as

$$c(x, y, 0) = Q_0 \exp\left(-\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2}\right), \tag{4}$$

where, Q_0 represents the amplitude of the bell curve, while (x_0, y_0) denotes the center of the curve, corresponding to the location of the source. The parameters σ_x and σ_y control the width of the curve in the x and y directions, respectively. These values were kept constant for all the data sets used to train and test the Neural Network. This was done to simplify the problem and reduce the number of input variables for the model. Future works might include the input of different initial condition and physical parameters into the model, so it results in a more general solution.

In an ideal scenario for modeling the release of a substance from a point source, setting σ_x and σ_y to zero would confine the initial concentration to a single point. However, this approach presents numerical challenges in the solution process. Even with a fine grid resolution, numerical errors caused by the singularity at the point source can result in instability and nonphysical oscillations in the solution. This compromises the accurate representation of the system's physical behavior.

To mitigate this issue, a Gaussian bell curve with small values of σ_x and σ_y is employed. This choice enables a more precise and stable solution. Figure 1 and Figure 2 visually depict an example of the initial condition, with parameters $Q_0 = 100$, $x_0 = y_0 = 1$, and $\sigma_x = \sigma_y = 0.03$.

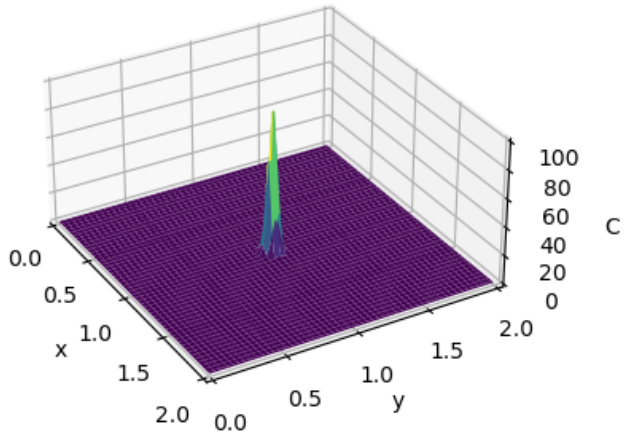

Figure 1. Initial condition curve.

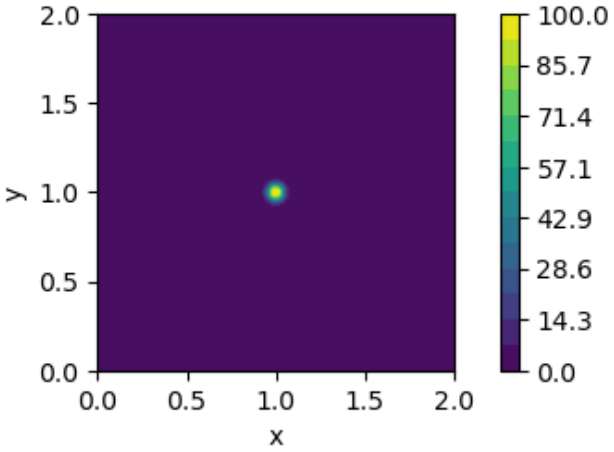

Figure 2. Contour plot of the initial condition curve.

2.2 Numerical Solution

With the definition of the partial differential equation, the initial and the boundary conditions, the bidimensional advection diffusion problem can now be solved using numerical methods. In this study, an implicit formulation of the Finite Difference Method (FDM) was employed. The implicit formulation ensures numerical stability, which results in accurate and reliable solutions. However, this approach comes at a cost of increased computational intensity (Noye and Tan, 1989; Silva Neto and Becceneri, 2009).

The implicit FDM involves discretizing the equations in space and time, resulting in a system of algebraic equations that can be solved iteratively. This process provides the numerical solution for different time steps, revealing the temporal evolution of the substance's concentration distribution.

To apply the FDM to the advection-diffusion equation, the domain is discretized into a grid of N_x equally spaced points in the x direction and N_y equally spaced points in the y direction, with grid spacings $\Delta x = \frac{L_x}{N_x - 1}$ and $\Delta y = \frac{L_y}{N_y - 1}$, respectively. This creates a total of $(N_x - 2) \cdot (N_y - 2)$ interior grid points, where the concentration values will be computed. The boundary grid points are treated separately, using the appropriate boundary conditions. Figure 3 shows an example of a discretized mesh with $N_x = N_y = 10$.

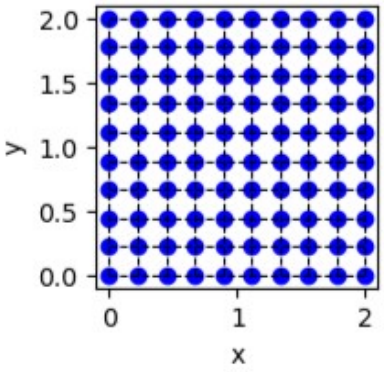

Figure 3. Example of discretized mesh.

The initial concentration distribution is then set on the interior grid points at time $t = 0$, and the finite difference equations are iteratively solved at each time step to evolve the concentration distribution in time. The derivative approximations for the implicit method are obtained by replacing the time derivative with a forward difference, the spatial derivatives of the diffusion term with a central difference and the derivatives of the advection term with backward differences, all evaluated at an advanced time instant. The resulting system of linear equations can be solved using matrix methods such as Gaussian elimination or iterative methods such as the Jacobi or Gauss-Seidel method.

The approximations for the derivatives used are as follows:

$$\frac{\partial c}{\partial t} \approx \frac{c_{i,j}^{n+1} - c_{i,j}^{n}}{\Delta t} \tag{5}$$

$$\frac{\partial^2 c}{\partial x^2} \approx \frac{c_{i+1,j}^{n+1} - 2c_{i,j}^{n+1} + c_{i-1,j}^{n+1}}{\Delta x^2} \tag{6}$$

$$\frac{\partial^2 c}{\partial y^2} \approx \frac{c_{i,j+1}^{n+1} - 2c_{i,j}^{n+1} + c_{i,j-1}^{n+1}}{\Delta y^2} \tag{7}$$

$$\frac{\partial c}{\partial x} \approx \frac{c_{i,j}^{n+1} - c_{i-1,j}^{n+1}}{\Delta x} \tag{8}$$

$$\frac{\partial c}{\partial y} \approx \frac{c_{i,j}^{n+1} - c_{i,j-1}^{n+1}}{\Delta y} \tag{9}$$

where the indexes $i = 0, 1, \ldots, N_x - 1$ and $j = 0, 1, \ldots, N_y - 1$ are used to represent the discrete position of a point on the grid on the x and y directions, respectively. The index n is used to represent the discrete time level of the solution, with $n = 0$ corresponding to the initial time and each subsequent value of n representing a later time step in the solution. Therefore, $c_{i,j}^{n}$ represents the concentration value at grid point (i, j) at time instant $n\Delta t$.

Substituting Eqs. (5) to (9) into Eq. (1), results at the following equation:

$$\begin{aligned} &\frac{c_{i,j}^{n+1} - c_{i,j}^{n}}{\Delta t} + u\frac{c_{i,j}^{n+1} - c_{i-1,j}^{n+1}}{\Delta x} + v\frac{c_{i,j}^{n+1} - c_{i,j-1}^{n+1}}{\Delta y} \\ &\qquad = D\left(\frac{c_{i+1,j}^{n+1} - 2c_{i,j}^{n+1} + c_{i-1,j}^{n+1}}{\Delta x^2} + \frac{c_{i,j+1}^{n+1} - 2c_{i,j}^{n+1} + c_{i,j-1}^{n+1}}{\Delta y^2}\right) \end{aligned} \tag{10}$$

that can be rearranged as follows

$$\begin{aligned} (-U_x - K_x)C_{i-1,j}^{n+1} + (-U_y - K_y)C_{i,j-1}^{n+1} + (1 + U_x + U_y + 2K_x + \\ + 2K_y)C_{i,j}^{n+1} + (-K_y)C_{i,j+1}^{n+1} + (-K_x)C_{i+1,j}^{n+1} = C_{i,j}^{n} \end{aligned} \tag{11}$$

This equation describes the concentration at the interior grid points of the discretized domain, where U_x, U_y, K_x and K_y represent coefficients obtained from the discretization of the partial derivatives of the advection and diffusion terms, respectively. They are given by:

$$U_x = \frac{u\Delta t}{\Delta x} \tag{12}$$

$$U_y = \frac{v\Delta t}{\Delta y} \tag{13}$$

$$K_x = \frac{D\Delta t}{\Delta x^2} \tag{14}$$

$$K_y = \frac{D\Delta t}{\Delta y^2} \tag{15}$$

The FDM solution was verified by comparing it with the Finite Element Method (FEM) solution, which is a reliable alternative method. The FEM solution was obtained using the NdSolve function of the Wolfram Mathematica 12.0 software. The NdSolve function uses adaptive mesh refinement and error control to ensure accuracy and efficiency. Figure 4 shows a comparison between the FDM and FEM solutions for one example case. These curves represent the concentration observed at the center of the domain, at $y = 1.0$, at time instant $t = 1.0$. To test the robustness of the FDM method, 30 different cases were simulated, with different mesh sizes and initial conditions. The Root Mean Squared Error between the FDM and FEM solutions for these cases was 5.56E-2. These results indicate that both numerical methods are in good agreement, which confirms the consistency and accuracy of the FDM solution.

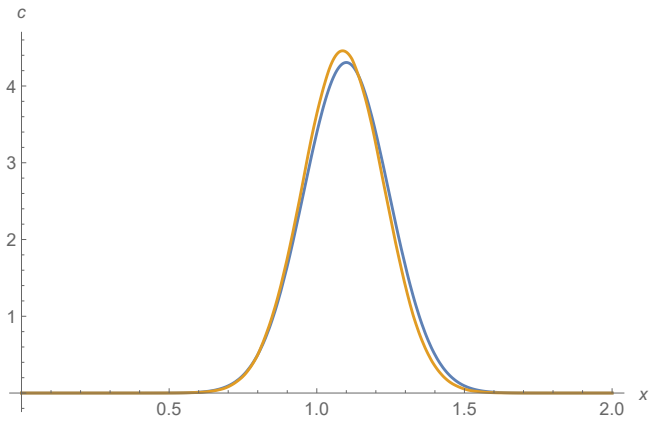

Figure 4. Comparison between different numerical solutions.

3. Neural Network Model

Artificial neural networks are systems consisted of interconnected processing nodes, often referred to as “neurons," that can process and transmit information. These networks can recognize complex patterns and make predictions by adjusting the connections between neurons. A widely used type of neural network is the Multilayer Perceptron (MLP) network. It consists of multiple layers of interconnected neurons, with each layer processing information from the previous layer. Starting with an input layer, the data is passed through each subsequent layer until the final output layer produces a prediction or classification (Gardner and Dorling, 1998; Shams *et al.*, 2021).

To train an MLP network, a set of input-output pairs is provided, and the training process is repeated for a specific number of iterations or epochs. During each epoch, the network adjusts its internal weights and biases to minimize the difference between predicted and actual outputs. In this particular problem, the input layer contains source coordinates, and the output layer consists of concentration values observed by a group of 12 sensors. The training process continues iteratively until the network can accurately predict sensor readings for new and unseen source locations. The MLP structure is chosen for its simplicity and effectiveness in solving regression problems (Bishop, 1994).

3.1 Training Data

The numerical solution presented in Section 2served as synthetic data for training an artificial neural network. By feeding the network with this data, the model can learn the underlying patterns and relationships within the system. This training process enables the neural network to subsequently predict and analyze the behavior of the substance under different conditions, offering a valuable tool for further exploration and analysis of the bidimensional advection diffusion problem.

The dataset used to train this model was generated by solving the bidimensional advection diffusion problem for 800 different source coordinates within the domain and collecting data at a set of predetermined sensor locations, at a specific time instant. The source coordinates are defined by varying the values of x_0 and y_0 in Eq. (4). For each case, the advection diffusion problem was solved for one step advanced in time, rendering the data collected at a time instant $t_1 = \Delta t$, where the time step was set to $\Delta t = 0.1$.

The 12 sensor locations are arranged in a semi-ellipse shape inside the domain, as shown on Fig. 5. The source coordinates will vary within the region in front of the sensors. This approach generates a comprehensive range of scenarios that can be used to train the model.

The physical parameters used to generate the training data were: velocity vector $\mathbf{v} = (1, 0)$, diffusion coefficient of $D = 0.01$, and domain size of $L_x = L_y = 2$. Data from the 800 different cases simulated were divided into three categories: 300 cases were used for training, 250 for model validation, and the other 250 cases were never presented to the model during training phase, and are used solely for testing.

3.2 Neural Network Model Architecture

The complexity of the problem and the amount of data available influence the choice of the number of hidden layers and neurons in each layer. A larger number of hidden layers and neurons may model more complex patterns but can also cause overfitting (Dombi and Jónás, 2022). The Sigmoid function, which is a common activation function in neural networks (Pratiwi *et al.*, 2020), was used as the activation function in this MLP model. The model used Adam as the optimizer, which is a stochastic gradient descent optimization algorithm that adjusts the learning rate for each weight based on the first and second moments of the gradients. This optimizer helps the model achieve faster and better convergence.

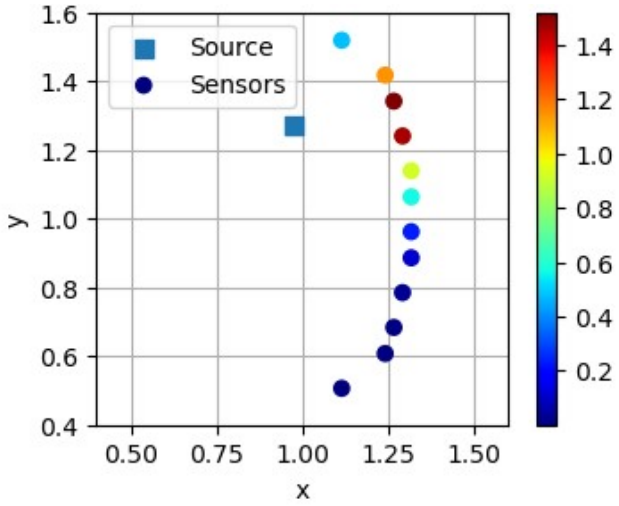

Figure 5. Sensors Network.

The model used Mean Squared Error (MSE) as the loss function, which is a frequently used loss function that computes the average squared difference between the predicted and actual values. MSE is appropriate for regression problems and helps the model learn how to minimize the difference between predicted and actual values, and is calculated using the following equation:

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{16}$$

where y is the actual value, $\hat{y}$ is the prediction, and n is the number of samples.

An early stopping technique was applied to stop the training process when the validation loss stopped decreasing. This technique helps to prevent overfitting and save computational resources. The early stopping criterion was set to monitor the validation loss and stop the training if it did not improve for 20 consecutive epochs. The model was then restored to the best state based on the validation loss. The early stopping technique was used for all the cases, and the optimal number of epochs varied depending on the convergence rate and the model's performance on the validation dataset.

To determine the number of hidden layers and number of neurons per layer, tests with different configurations were conducted. Table 1 presents the results obtained by the different layer structures.

Table 1. Results obtained for different MLP configurations

# of Hidden Layers	Nodes per Layer	MSE ($\times 10^{-3}$)		
		Training	Validation	Test
2	24	4.856	4.927	5.201
	32	4.358	4.427	5.339
	48	3.790	3.714	4.506
3	24	3.120	3.323	3.921
	32	**2.959**	**2.850**	**3.850**
	48	2.842	2.847	4.364
5	24	2.563	2.791	4.481
	32	1.842	2.012	4.240
	48	1.613	1.923	4.402

Based on the observed results of the MSE of the Testing dataset, the best model has 3 hidden layers with 32 nodes, with a test error of 3.850×10^{-3}. Using more than 3 hidden layers may lead to overfitting, as evidenced by the 5 hidden layers' significant increase in test error. Therefore, the 3 hidden layers with 32 nodes per layer strike a good balance between model complexity and generalization, making it the chosen option.

Model configuration is highly problem-dependent, and different problems may require different choices. In future work, further experimentation and optimization could be pursued to fine-tune the model architecture and configuration, considering specific characteristics of the problem domain.

In terms of computational efficiency, the ANN model generated predictions for all 800 cases in less than a minute on

a computer with a CPU clocked at 3.5 GHz and 16 GB of RAM. This time is based on multiple runs to ensure accuracy and applies to the generation of all 800 different cases. In comparison, the FDM approach took about 27 minutes to generate predictions for the same number of cases. The significant difference in the generation time of these two methods highlights the potential advantage of the proposed model in terms of computational efficiency.

4. INVERSE PROBLEM

Inverse problems aim to identify the parameters of a model that best explain the observed data. In this problem, the source parameters are the coordinates of the pollutant or gas release, and the observed data are the concentration measurements at different points in space, calculated by the numerical approach presented in Section 2.1. The inverse problem algorithm can be formulated as an optimization problem, where the objective is to minimize the discrepancy between the predicted data and the observed data, subject to some constraints on the source parameters.

The function to be optimized can be written as:

$$\mathcal{F}_{\text{obj}}\left(\mathbf{Z}\right) = \sum_{i=1}^{N_{\text{s}}} \left(C_{\text{NN}}^{i}(\mathbf{Z}) - C_{\text{obs}}^{i}\right)^{2} \tag{17}$$

where $\mathcal{F}_{\text{obj}}$ is the objective function that measures the discrepancy between the predicted and observed concentrations. $\mathbf{Z}$ is the vector of source coordinates. N_{s} is the number of sensors where the concentration measurements are taken. C_{NN}^{i} is the predicted concentration at the i-th sampling point, obtained by using the MLP network as a forward operator, and C_{obs}^{i} is the observed concentration at the same location. The objective function is minimized when the predicted and observed concentrations are close to each other for all sampling points.

The Luus-Jaakola (LJ) method served as the chosen algorithm for solving this inverse problem. The LJ method is a heuristic optimization technique designed for global optimization of a real-valued function. Unlike methods that require the computation of function gradients, the LJ method evaluates the objective function at different points within the parameter space. This is accomplished through the utilization of uniform random sampling from a neighborhood around the current position. Throughout the iterations, the size of the neighborhood diminishes, ultimately leading to a convergence of iterates towards a cluster point.

The LJ method is suitable for solving our inverse problem because it can handle complex and nonlinear forward operators without requiring their explicit formulation or differentiation. Moreover, it can deal with non-convex and non-smooth objective functions, which may arise due to noise or uncertainty in the observed data. The LJ method is also easy to implement and has few parameters to tune. However, the LJ method may suffer from slow convergence or premature termination, especially when the parameter space is large or high-dimensional.

5. RESULTS

The performance of the inverse problem algorithm was evaluated by considering 40 different source locations in the domain, uniformly distributed over a grid of 8 by 5 points. For each source location, synthetic data was generated by solving the advection diffusion equation with the Implicit Finite Difference method, and then the inverse problem algorithm was applied to estimate the source parameters using the MLP network as a forward operator. This estimation process was repeated 15 times for each source location, using different initial points and random vectors, to ensure the accuracy and robustness of the results, and the best values were considered for the results presented. This resulted in a total of 600 simulations for the inverse problem. It was found that using the MLP network as a surrogate for the forward operator was a good approach, as it significantly reduced the computational time required to solve the inverse problem. The MLP network was able to simulate the direct problem about 27 times faster than the numerical approach, on average, which made the inverse problem algorithm more efficient and feasible.

The results of the inverse problem algorithm were quantified by calculating the Euclidean distance between the predicted and exact source coordinates for each of the 40 cases. The Euclidean distance is given by

$$d = \sqrt{(x_{\text{p}} - x_{\text{e}})^2 + (y_{\text{p}} - y_{\text{e}})^2}, \tag{18}$$

where $(x_{\text{p}}, y_{\text{p}})$ and $(x_{\text{e}}, y_{\text{e}})$ are the predicted and exact source coordinates, respectively. The average distance, the median distance and the standard deviation are shown in Table 2.

The results show that the inverse problem algorithm was able to estimate the source location with high accuracy and low variability, as the RMSE, median distance and standard deviation of distance were all small compared to the size of the domain. Figure 6 illustrates the median distance case, which is a good visual representation of the accuracy of the model.

Table 2. Statistics of the distances between predicted and exact source coordinates.

Statistic	**Value**
Average distance	0.01073
Median distance	0.00794
Standard deviation	0.00726

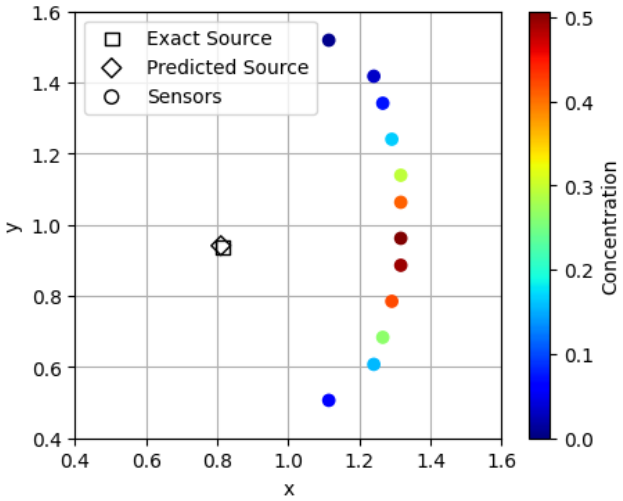

Figure 6. The median distance case, comparing the predicted and exact source coordinates.

6. CONCLUSION

This paper addressed the problem of identifying pollutant sources through the application of inverse modeling techniques to predict pollutant behavior in the atmosphere. An artificial neural network (ANN) was used as the direct problem solution, trained with synthetic data obtained by numerically solving the bidimensional advection diffusion equation.

The research demonstrated the effectiveness of the MLP network in accurately predicting concentration values at sensor locations. This facilitated the development of a computationally efficient solution for the inverse problem, which aimed to identify source parameters based on observed concentration data. The Luus-Jaakola method, a heuristic optimization algorithm, successfully minimized the discrepancy between predicted and observed concentrations, enabling estimation of source coordinates.

By combining the data obtained by the advection diffusion model, the MLP network, and the Luus-Jaakola algorithm, this study showcased a feasible and efficient approach to identifying pollutant sources. The implications of this approach are significant for environmental monitoring and emergency response systems, as it allows for targeted actions to mitigate pollutants or furtive gas leaks.

Future research can explore alternative neural network architectures or optimization algorithms to further enhance the efficiency and accuracy of the inverse modeling process. Additionally, incorporating real-world data and considering more complex scenarios can improve the applicability and robustness of the developed approach.

7. ACKNOWLEDGEMENTS

The authors acknowledge the financial support provided by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, Finance Code 001), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ). This work was conducted during a scholarship supported by the International Cooperation Program CAPES PrInt at the University of Granada. Financed by CAPES – Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Ministry of Education of Brazil with proccess number 88887.716424/2022-00.

D. Pelta acknowledges support from projects PID2020-112754GB-I00, MCIN /AEI/10.13039/501100011033 and FEDER/Junta de Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades/Proyecto (BTIC-640-UGR20).

8. REFERENCES

Bishop, C.M., 1994. "Neural networks and their applications". *Review of scientific instruments*, Vol. 65, No. 6, pp. 1803–1832.

Dombi, J. and Jónás, T., 2022. "The generalized sigmoid function and its connection with logical operators". *Int. J. of Approximate Reasoning*, Vol. 143, pp. 121–138.

Gardner, M.W. and Dorling, S., 1998. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences". *Atmospheric environment*, Vol. 32, No. 14-15, pp. 2627–2636.

Liu, F., Zhuang, P., Anh, V. and Turner, I., 2007. "Stability and convergence of the diff. methods for the space–time fractional adv.–diff. eq." *App. Mathematics and Comp.*

Luo, X., Bing, H., Luo, Z., Wang, Y. and Jin, L., 2019. "Impacts of atmospheric particulate matter pollution on environmental biogeochemistry of trace metals in soil-plant system: A review". *Environmental Pollution*, Vol. 255, p. 113138. ISSN 0269-7491. doi:https://doi.org/10.1016/j.envpol.2019.113138. URL `https://www.sciencedirect.com/science/article/pii/S0269749119319864`.

Moreira, D., Vilhena, M. and Buske, D., 2009. "The state-of-art of the giltt method to simulate pollutant disp. in the atmosph." *Atmospheric Research*, Vol. 92, No. 1, pp. 1–17.

Noye, B. and Tan, H., 1989. "Fin. diff. methods for solving the two-dimensional advection-diffusion equation". *Intl. Journal for Numerical Methods in Fluids*, Vol. 9, No. 1, pp. 75–98.

Pratiwi, H., Windarto, A.P., Susliansyah, S., Aria, R.R., Susilowati, S., Rahayu, L.K., Fitriani, Y., Merdekawati, A. and Rahadjeng, I.R., 2020. "Sigmoid activation function in selecting the best model of artificial neural networks". *Journal of Physics: Conference Series*, Vol. 1471, No. 1, p. 012010. doi:10.1088/1742-6596/1471/1/012010. URL `https://dx.doi.org/10.1088/1742-6596/1471/1/012010`.

Prieto, F.U., Muñoz, J.J.B. and Corvinos, L.G., 2011. "Application of the generalized fin. diff. method to solve the adv.–diff. eq." *Journal of Comp. and App. Math.*

Shams, S.R., Jahani, A., Kalantary, S., Moeinaddini, M. and Khorasani, N., 2021. "The evaluation on artificial neural networks (ann) and multiple linear regressions (mlr) models for predicting so2 concentration". *Urban Climate*, Vol. 37, p. 100837. ISSN 2212-0955. doi:https://doi.org/10.1016/j.uclim.2021.100837. URL `https://www.sciencedirect.com/science/article/pii/S2212095521000675`.

Silva Neto, A.J. and Becceneri, J.C., 2009. "Técnicas de inteligência computacional inspiradas na natureza–aplicação em problemas inversos em transferência radiativa". *Soc. Brasileira de Mat. Aplicada e Comput. (SBMAC)*, Vol. 41, p. 122.

Stockie, J.M., 2011. "The math. of atmosph. dispersion modeling". *Siam Review*.

9. RESPONSIBILITY NOTICE

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