Modeling of Mexican Hat Wavelet Neural Network with L-BFGS Algorithm for Simulating the Recycling Procedure of Waste Plastic in Ocean

Arup Kumar Sahoo , Snehashish Chakraverty[*]

Department of Mathematics, National Institute of Technology Rourkela, Rourkela 769008, India

[*] Correspondence: Snehashish Chakraverty (sne_chak@yahoo.com)

Abstract: In the global economy, plastics are considered a versatile and ubiquitous material. It can reach to marine ecosystems through diverse channels, such as road runoff, wastewater pathways, and improper waste management. Therefore, rapid mitigation and reduction are required for this ever-growing problem. The marine habitats are believed to be the highest emitters and absorbers of O_2 and CO_2 respectively. As such, every day, the prominence of managing the litter in the ocean is growing effectively and efficiently. One of the most significant challenges in oceanography is creating a comprehensive meshless algorithm to handle the mathematical representation of waste plastic management in the ocean. This research dedicates to studying the dynamics of waste plastic management model governed by a mathematical representation depending on three components viz. Waste plastic (W), Marine litter (M) and Recycling of debris (R), i.e., WMR model. In this regard, an unsupervised machine learning approach, namely Mexican Hat Wavelet Neural Network (M_hWNN) refined by the efficient Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS), i.e., M_hWNN-LBFGS model has been implemented for handling the non-linear phenomena of WMR models. Besides, the obtained solution is meshfree and compared with the state-of-art numerical result to establish the precision of the M_hWNN-LBFGS model. Furthermore, different global statistical measures (MAPE, TIC, RMSE, and ENSE) have been computed at twenty testing points to validate the stability of the proposed algorithm.

Keywords: Wavelet neural network; Mexican hat wavelet; Meshless; Waste plastic management; WMR model; Unsupervised

1. Introduction

Plastic contamination has spread across a wide swath of the ocean due to its lightness and solidness properties. From zooplankton to cetaceans, marine mega fauna suffers a direct and fatal consequence of plastic contamination. Every year, thousands of seabirds, seals, turtles and other marine reptiles are being killed through ingestion, entrapment and being entangled in plastic. The lower trophic-level lives and their predators in aquatic environments are being affected by consuming diligent natural toxins adhering to plastic. The presence of drifting plastics, extending from huge abandoned nets, docks and cruises that carry fish, green growth, and microbial networks to non-local districts, further exacerbates these consequences [1]. So, the management of waste plastic in the ocean has received increased attention in recent years from researchers around the world as well as from different activists and government bodies. Ocean plastic pollution was specifically mentioned in high-level agreements like the Berlin declaration in 2013 and the resolution of G7 Leaders in 2015 [2]. EU legislation also passed an amendment, notably the *Marine Strategy Framework Directive*, which helped move this issue up the international agenda [3]. Despite different oceanographic models, there is no *one size fits all* methodology to waste management of the ocean. To reduce, recycle, and clean up waste plastic, different strategies have been used around the world. When these strategies are supplemented by the science-driven mathematical model, they may be most effective. However, machine learning (ML), in particular neural techniques, has shown its potential by

surpassing human levels of accuracy for simulating complex phenomena related to oceanography.

Artificial intelligence (AI) has been a subject of intense media hype in the 21st century. Different branches of AI i.e. machine learning, deep learning, and ANN come up in many articles, irrespective of field. A wide-scale application of various phenomena in machine translation, non-linear pattern recognition, medical diagnosis, image processing, robotics, and speech & face detection, are well described by ANN [4, 5]. As ANN becomes widespread and integrated with human-centric applications and algorithms, so the focus has returned to explainability. Over the last two decades, the implementation of ANN has sprung up for solving different types of differential, integral, and algebraic equations. It is often characterized as being a black box. That is, the closed-form of neural solution is available to predict the value at any testing point of the given domain.

Although NN has the universal approximation power, it still has some shortcomings as it fails to define the local features such as jumps in the objective function, discontinuities in curvature, local minima and slow learning rate [6]. As such, an alternative neural network model based on the combination of some particular wavelet kernels and feed-forward neural networks, namely, the wavelet neural networks (WNN), has been proposed. It seems to be an effective and strong approximate model for universal functions. Additionally, the learning rate of the WNN is relatively faster than conventional NN.

Optimizers are algorithms or methods used to minimize the loss function. It changes the attributes of a neural network, such as weights and biases. A good number of optimization algorithms have emerged during the last few years as remarkable advances both in application areas and research. It can be classified as derivative-based or derivative-free. The most common technique for optimizing a function is using derivative-based algorithms. In this regard, some potential derivative-based optimization algorithms are gradient descent (GD), conjugate gradient (CG), stochastic gradient descent (SGD) and Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). The derivative-based optimization algorithms are more stable when compared to the derivative-free optimization algorithms [7].

The objective of this article is to illustrate the Mexican Hat Wavelet Neural Network with L-BFGS optimization algorithm for simulating the recycling procedure of waste plastic in the Ocean. The non-linear form of waste plastic management model of the Ocean is represented by three elements: Waste plastic (W), Marine litter (M) and Recycling of debris (R), i.e., WMR model. In this regard, various instances have been discussed, and neural predictions have been made at testing points. As the proposed neural method is meshfree so after training of the network, we can find the solution at any point inside the given domain of the DE.

The rest of our contribution can be outlined as follows: in **Section 2**, a literature review is presented. **Section 3** presents the preliminaries of WMR model and architecture of M_hWNN for the sake of completeness. In **Section 4**, the formulation of M_hWNN-LBFGS algorithm to solve WMR model is discussed. In **Section 5**, two cases of WMR model have been investigated to verify the effectiveness of the M_hWNN -LBFGS model. Finally, the conclusions are drawn in **Section 6**.

2. Related Studies

In 1943, Warren S. McCulloch and Walter Pits proposed neuron activities that merged the studies of neurophysiology and mathematical logic. In their historical paper [8], they developed the first elementary model of ANN, in which neurons were escorted by the "all-or-none" process. It brings revolution in the field of AI and attracts researchers to work on it.

In a pioneering work, Alizadeh and Kavianpour [9] developed a wavelet-ANN model for accurate predictions of dissolved oxygen, temperature and salinity in the Pacific Ocean. Chen et al. deployed a pre-clustering ANN model, using different components of sea-surface wind speed to estimate the mixed-layer depth in the Indian Ocean [10]. For speed and accurate detection of deep-sea debris, Xue et al. proposed a deep NN model [11]. Nuwairan et al. [12] developed a supervised neural network to simulate the waste plastic management model of ocean. Motivated by the above considerations, it is natural to propose a new and efficient ANN algorithm to understand the dynamical behaviour of waste plastics management in ocean. In this work, we have designed an advanced neural model by combining the properties of mexican hat wavelet basic along with the L-BFGS training algorithm, viz. M_hWNN-LBFGS for the study of non-linear phenomena of the WMR model.

ANN has been convincingly used in the field of differential equation (DE) in the past couple of years after Lee and Kang [13] developed a novel Hopfield neural network model to find the solution of first-order DE. In 1998, Lagaris et al. [14] employed the concept of an unconstrained optimization problem and proposed a trial solution for DE with regular boundaries that satisfies the given boundary and initial conditions. However, some researchers have developed different ODE and PDE solvers for DE with specific properties. For instance, Lie symmetry differential equations [15], fractional differential equations [16], fuzzy differential equations [17], and singularly perturbed differential equations [18]. In another approach, symplectic artificial neural network model using curriculum learning has been investigated by Sahoo and Chakraverty [19]. In this regard, a few more potential models are the spherical gas cloud model by Ahmad et al. [20], Legendre artificial neural network method by Verma and Kumar [16], etc. Nonetheless, these methods are associated with problems with well-defined boundaries.

As per the literature, the advantages of NN are further strengthened with the addition of some particular types of wavelets. As it overcomes the drawbacks of NN and converts it into an efficient technique for universal function approximation. WNN has been extensively used by researchers and scientists in different fields such as fault diagnosis [21], daily pan evaporation [22], energy price forecasting [23], industrial robot manipulators [24], Adaptive PID controller design [25], etc.

In order to explore the solution of fractional differential equations, Sabir et al. introduced a fractional Meyer wavelet neural network model to solve nonlinear singular fractional Lane–Emden systems [26]. Tan et al. [6] investigated solutions of PDE using an unsupervised WNN model with meta-heuristic algorithm. Wu et al. [27] proposed a WNN with the structure 1×N×1 to find the numerical solution of fractional differential equations. These literatures stimulate the authors to investigate different wavelet kernels as an alternative, reliable, efficient, and robust computing paradigm for solving non-linear phenomena of the oceanographic WMR model.

The innovative insights of the M_hWNN-LBFGS model are summarized as below:

- A novel multilayer framework namely Mexican Hat Wavelet Neural Network has been designed under the Jupyter notebook environment.
- Neural simulation of the non-linear waste plastic management model of the Ocean i.e., WMR model has been demonstrated by using the proposed algorithm.
- The resilience of M_hWNN-LBFGS model is observed by comparing the obtained simulation results with RK4 method.
- In addition, different global statistical measures (MAPE, TIC, RMSE, and ENSE) have been calculated at testing points to validate the stability of the proposed algorithm.
- An unsupervised training algorithm ensures that the M_hWNN-LBFGS is a powerful tool to predict the solution of any other non-linear system of equations.

3. Preliminaries

In this section, an overview of the waste plastic management WMR model and architecture of the multilayer Mexican Hat Wavelet Neural Network has been discussed.

3.1 Overview of WMR Model

The WMR model is represented via three components, i.e., Waste plastic $W(\gamma)$ Marine litter $M(\gamma)$ and Recycling of debris $R(\gamma)$ which construct a non-linear WMR system as shown below [12]:

$$
\begin{cases}
W'(\gamma) = \alpha R(\gamma) - \beta W(\gamma) - \eta M(\gamma) W(\gamma) + \overline{b}, & W(a_1) = \lambda_1, \\
M'(\gamma) = \eta M(\gamma) W(\gamma) - \delta M(\gamma), & M(a_2) = \lambda_2, \\
R'(\gamma) = \beta W(\gamma) + \delta M(\gamma) - (\alpha + \theta) R(\gamma). & R(a_3) = \lambda_3,
\end{cases}
\tag{1}
$$

where, α is the rate of recycled waste to regenerate new waste, β is the rate of waste to be recycled directly, η denotes the rate of waste to enter into the marine, $\overline{b}$ is the new waste rate to be reproduced, δ is the marine litter rate to recycle, and θ stands for the recycled waste rate to be lost.

3.2 Architecture of Mexican Hat Wavelet Neural Network (M_hWNN)

ANN is a branch of artificial intelligence (AI) that mimics the training process of the human brain to predict patterns from given historical data. Neural networks are processing devices of mathematical algorithms that can be implemented by computer languages.

Wavelet is a 'small wave' function written as $\psi(\gamma) \in L^2(R)$ with the property,

$$
\int_{-\infty}^{\infty} \psi(\gamma) d\gamma = 0
\tag{2}
$$

and centred in the neighbourhood of 0. Wavelet transform has time-frequency localization property where as NN has self-adaptive, fault tolerance, robustness, and strong inference ability. The network topology of a wavelet neural network is very similar to a feed-forward multi-layer neural network. The hidden layer consists of wavelet neurons commonly known as wavelons, whose activation functions are drawn from a wavelet basis. In accordance with some learning algorithms, the translation and dilation of the wavelets along with the weights are updated.

The mother wavelet is made up of two factors viz. translation and the dilation c_i, where i denotes the ith wavelet [28] and is written as:

$$\Psi(\gamma) = \frac{1}{\sqrt{|c_i|}} \psi\left(\frac{\gamma - d}{c}\right), c > 0, d \in R \tag{3}$$

As such, the following theorem is stated in the literature regarding the properties and rate of convergence of WNN.

Theorem 1. The WNN has L^2 and universal function approximation properties [29].

Proof of Theorem 1. For the proof of Theorem 1 see Ref. [29] by Zhang et al.

In this work, the Mexican Hat mother wavelet function is used as wavelet basis which can be denoted as

$$\psi_{Mexicanhat} = (1 - \gamma)^2 \, e^{-\gamma^2/2}. \tag{4}$$

Mexican Hat mother wavelet is obtained from Gaussian function by applying the Laplacian operator and it is a continuously differentiable wavelet. Figure 1 shows graphical representation of the Mexican Hat wavelet.

The output of M_hWNN is defined as

$$wnn(\gamma, \vec{\upsilon}) = \sum_{i=1}^{n} \mu_{i,j} \psi_i(\gamma_1, \gamma_2, ..., \gamma_N) + b_j, \, j = 1, 2, ... k \tag{5}$$

where, $\mu_{i,j}$ is the weights from the input unit i to the hidden unit j, b_j is the bias and $\vec{\upsilon}$ is the trainable parameters.
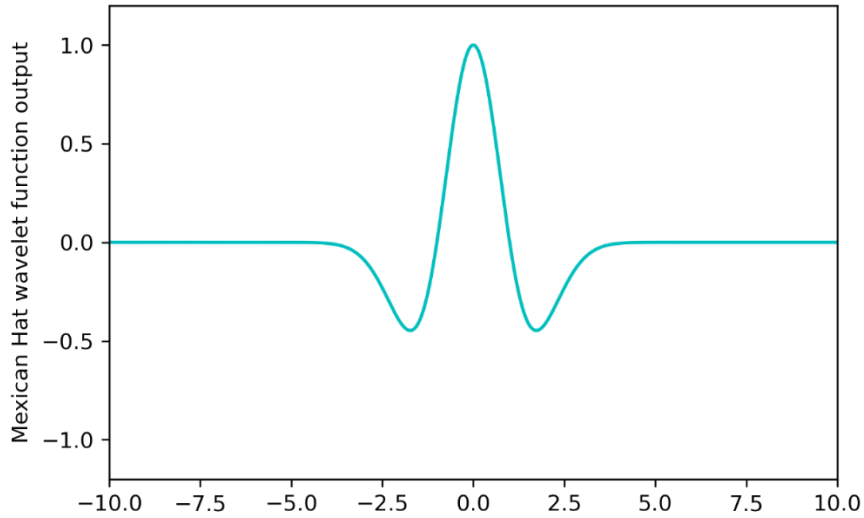

Figure 1. Mexican hat mother wavelet function

4. Modeling of Mexican Hat Wavelet Neural Network for WMR Model

This section explains the formation of the proposed M_hWNN-LBFGS technique to solve the WMR model as an unconstrained problem.

An autonomous system of WMR model can be written in matrix form as:

$$\begin{pmatrix} f_1\left(\gamma, W, \dfrac{dW}{d\gamma}\right) \\ f_2\left(\gamma, M, \dfrac{dM}{d\gamma}\right) \\ f_3\left(\gamma, R, \dfrac{dR}{d\gamma}\right) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{6}$$

First the governing equation (Eq. (1)) will be transferred into an approximate solution, as a combination of initial/boundary conditions, a user-defined mathematical expression, and M_hWNN-LBFGS output. Therefore, we have

$$
\begin{cases}
\tilde{W}\left(\gamma,\vec{\upsilon}_w\right)=\lambda_1+(1-e^{-(\gamma)})\,wnn\left(\gamma,\vec{\upsilon}_w\right), \\
\tilde{M}\left(\gamma,\vec{\upsilon}_m\right)=\lambda_2+(1-e^{-(\gamma)})\,wnn\left(\gamma,\vec{\upsilon}_m\right), \\
\tilde{R}\left(\gamma,\vec{\upsilon}_r\right)=\lambda_3+(1-e^{-(\gamma)})\,wnn\left(\gamma,\vec{\upsilon}_r\right),
\end{cases}
\tag{7}
$$

where, the first term $\lambda_i, i=1,2,3$ satisfies the initial condition of the given WMR model without adjustable parameters and second term is the neural output. For the given input $\gamma \in R^n$, the unknown function $wnn(\gamma,\vec{\upsilon}_t)$ in the second part of approximate solution is denoted by

$$
\begin{cases}
wnn\left(\gamma,\vec{\upsilon}_w\right)=\left(\displaystyle\sum_{j=1}^{k} w_j^w\left(1-z_j\right)^2 \exp\left(-z_j^2/2\right)\right), \\
wnn\left(\gamma,\vec{\upsilon}_m\right)=\left(\displaystyle\sum_{j=1}^{k} w_j^m\left(1-z_j\right)^2 \exp\left(-z_j^2/2\right)\right), \\
wnn\left(\gamma,\vec{\upsilon}_r\right)=\left(\displaystyle\sum_{j=1}^{k} w_j^r\left(1-z_j\right)^2 \exp\left(-z_j^2/2\right)\right),
\end{cases}
\tag{8}
$$

where, $z_j = \sum_{i=1}^{n} \mu_{i,j}\gamma_i + b_j, \mu_{i,j}$ and $w_j^x, x = w, m, r$ represent the weights from the input unit i to the hidden unit j and the hidden unit j to the output unit respectively, b_j is the bias and k denotes the number of neurons.

Let us denote the output of the third hidden layer as

$$
wnn\left(\gamma,\vec{\upsilon}\right)=O(\gamma_i),
\tag{9}
$$

where, $O(\gamma_i) = [o_1(\gamma_i), o_2(\gamma_i),\ldots,o_k(\gamma_i)]^T \in R^{k\times 1}$. Then $O(\gamma_i)$ can be obtained from

$$
\begin{bmatrix}
\psi(\mu_{1,1}\gamma_1+b_1) & \psi(\mu_{1,2}\gamma_1+b_2) & \ldots & \psi(\mu_{1,n}\gamma_1+b_n) \\
\psi(\mu_{2,1}\gamma_2+b_1) & \psi(\mu_{2,2}\gamma_2+b_2) & \ldots & \psi(\mu_{2,n}\gamma_2+b_n) \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\psi(\mu_{k,1}\gamma_k+b_1) & \psi(\mu_{k,2}\gamma_k+b_2) & \ldots & \psi(\mu_{k,n}\gamma_k+b_n)
\end{bmatrix}
\times
\begin{bmatrix}
\omega_1 \\ \omega_2 \\ \cdot \\ \cdot \\ \cdot \\ \omega_n
\end{bmatrix}
=
\begin{bmatrix}
o_1(\gamma_i) \\ o_2(\gamma_i) \\ \cdot \\ \cdot \\ \cdot \\ o_k(\gamma_i)
\end{bmatrix}.
\tag{10}
$$

Let us take the output matrix of the last hidden layer A and the weight vector ω as follows:

$$
A=
\begin{bmatrix}
\psi(\mu_{1,1}\gamma_1+b_1) & \psi(\mu_{1,2}\gamma_1+b_2) & \ldots & \psi(\mu_{1,n}\gamma_1+b_n) \\
\psi(\mu_{2,1}\gamma_2+b_1) & \psi(\mu_{2,2}\gamma_2+b_2) & \ldots & \psi(\mu_{2,n}\gamma_2+b_n) \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\psi(\mu_{k,1}\gamma_k+b_1) & \psi(\mu_{k,2}\gamma_k+b_2) & \ldots & \psi(\mu_{k,n}\gamma_k+b_n)
\end{bmatrix},
\quad
\omega=
\begin{bmatrix}
\omega_1 \\ \omega_2 \\ \cdot \\ \cdot \\ \cdot \\ \omega_n
\end{bmatrix}
\in R^{n\times 1}.
$$

and

Now define the block matrix E of the following form

$$E(\gamma) = \begin{bmatrix} A_w(\gamma) & 0 & 0 \\ 0 & A_m(\gamma) & 0 \\ 0 & 0 & A_r(\gamma) \end{bmatrix}$$

$$W = \begin{bmatrix} \omega_j^w & \omega_j^m & \omega_j^r \end{bmatrix}^T, \, j = 1,2...n$$

$$O = \begin{bmatrix} wnn(\gamma, \vec{\upsilon}_w) & wnn(\gamma, \vec{\upsilon}_m) & wnn(\gamma, \vec{\upsilon}_r) \end{bmatrix}^T$$

Then the above matrix can be compactly written as

$$E(\gamma) W = O \tag{11}$$

It may be noted that as the trainable parameters such as, weights and biases are fixed arbitrarily generated values, so $E(\gamma)$ depends only on training points γ.

In the next step, a training algorithm is employed to tune the adjustable parameters of M_hWNN-LBFGS, which is embedded in the approximate solution. The M_hWNN-LBFGS is trained to predict the solutions of WMR for any testing point inside the given domain by unsupervised learning, where the parameters are updated to minimise the objective function. In order to, find the objective function, we may need to calculate the gradient of the network $wnn(\gamma, \vec{\upsilon})$, that can be computed as follows:

$$D^\delta(wnn(\gamma, \vec{\upsilon})) = \frac{\partial^\delta wnn(\gamma, \vec{\upsilon})}{\partial \gamma^\delta} = \sum_{j=1}^k \omega_j \frac{\partial^\delta \psi}{\partial \gamma^\delta} = \sum_{j=1}^k \omega_j \, \mu_{i,j}^\delta \psi^{(\delta)}(z_j), \tag{12}$$

where, δ is order of derivative. Differentiate Eq. (7)

$$D^\delta \left[\tilde{W}(\gamma, \vec{\upsilon}_w), \tilde{M}(\gamma, \vec{\upsilon}_m), \tilde{R}(\gamma, \vec{\upsilon}_r) \right]$$

$$= D^\delta \left[(1-e^{-(\gamma)}) wnn(\gamma, \vec{\upsilon}_w), (1-e^{-(\gamma)}) wnn(\gamma, \vec{\upsilon}_m), (1-e^{-(\gamma)}) wnn(\gamma, \vec{\upsilon}_r) \right]$$

$$= D^\delta \begin{bmatrix} (1-e^{-(\gamma)}) \sum_{j=1}^k \sum_{i=1}^n \omega_j^w \left(1 - (\mu_{i,j}\gamma_i + b_j)\right)^2 \exp\left(\frac{-(\mu_{i,j}\gamma_i + b_j)^2}{2}\right), \\ (1-e^{-(\gamma)}) \sum_{j=1}^k \sum_{i=1}^n \omega_j^m \left(1 - (\mu_{i,j}\gamma_i + b_j)\right)^2 \exp\left(\frac{-(\mu_{i,j}\gamma_i + b_j)^2}{2}\right), \\ (1-e^{-(\gamma)}) \sum_{j=1}^k \sum_{i=1}^n \omega_j^r \left(1 - (\mu_{i,j}\gamma_i + b_j)\right)^2 \exp\left(\frac{-(\mu_{i,j}\gamma_i + b_j)^2}{2}\right) \end{bmatrix} \tag{13}$$

By using the gradient of the approximate solutions and for the given problem, the objective function can be formulated as follows:

$$J(\vec{\upsilon}) = \begin{cases} \sum_{i=1}^n \left(\frac{D\tilde{W}(\gamma, \vec{\upsilon}_w)}{D\gamma} - f_1(\gamma_i, \tilde{W}(\gamma, \vec{\upsilon}_w)) \right)^2 \\ \sum_{i=1}^n \left(\frac{D\tilde{M}(\gamma, \vec{\upsilon}_m)}{D\gamma} - f_2(\gamma_i, \tilde{M}(\gamma, \vec{\upsilon}_m)) \right)^2 \\ \sum_{i=1}^n \left(\frac{D\tilde{R}(\gamma, \vec{\upsilon}_r)}{D\gamma} - f_3(\gamma_i, \tilde{R}(\gamma, \vec{\upsilon}_r)) \right)^2 \end{cases} \tag{14}$$

On the other hand, we uniformly adopt L-BFGS as optimizer with a learning rate 0.01 to find the optimal

parameter. L-BFGS is a potential optimization technique based from the Quasi-Newton family that is widely employed in the field of deep learning. L-BFGS is same as BFGS except for the hessian matrix updation. As the training methodologies of the neural network are often iterative; so, we need to designate a starting point for iterations. Therefore, in our investigation, initial weights have been generated randomly and set to small numbers in [-1,1]\{0}. The graphical abstract of M_hWNN-LBFGS for solving WMR model is presented in Figure 2.
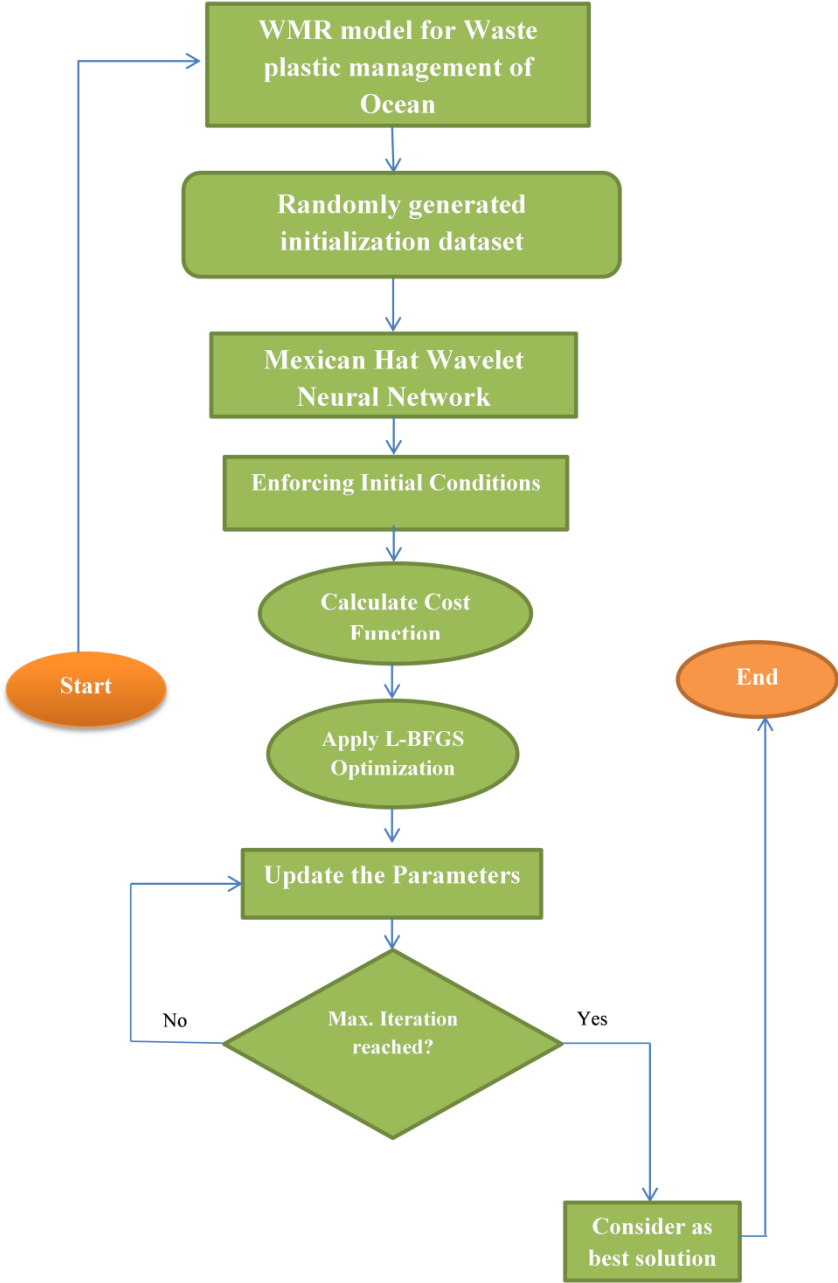
Figure 2. Framework of unsupervised M_hWNN-LBFGS algorithm for solving the WMR model

5. Simulation Results and Discussions

In order to manifest that the presented MhWNN-LBFGS algorithm is promising, we have addressed two problems for simulations in this section. The performances have been studied in terms of statistical measures between present results and traditional numerical results. All of the neural results in the following examples are implemented in the Jupyter notebook environment using Python 3.0. For both cases, authors have trained the

network for 1,000 epochs. However, after selecting the basic framework, hyperparameter tuning has been done to select the optimal number of hidden layers and the number of nodes in each hidden layer. The accuracy of the proposed M_hWNN-LBFGS algorithm has been shown in the tables and graphs. Different global statistical measures (NSE, MAPE, TIC, and RMSE) are evaluated, for convergence analysis of M_hWNN-LBFGS model which is defined as below [30]

$$
\text{NSE} = \left\{ 1 - \frac{\sum_{i=1}^{N}(\gamma_i - \hat{\gamma}_i)^2}{\sum_{i=1}^{N}(\gamma_i - \bar{\gamma}_i)^2}, \quad \bar{\gamma}_i = \frac{1}{N}\sum_{i=1}^{N}(\gamma_i) \right.
$$

$$
\text{ENSE} = 1 - \text{NSE}
$$

$$
\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\gamma_i - \hat{\gamma}_i}{\gamma_i}\right|
$$

$$
\text{TIC} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\gamma_i - \hat{\gamma}_i)^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\gamma_i)^2} + \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\gamma}_i)^2}}
$$

Problem 1 Here we have considered a non-linear WMR model by substituting the values of α=0.4, β=0.21, η=0.75, $\bar{b}$=0.36, δ=0.5, θ=0.05, a_i=0, λ_1=2, λ_2=1.5, and λ_3=1 in Eq. (1) [12]

$$
\begin{cases}
W'(\gamma) = 0.4\,R(\gamma) - 0.21W(\gamma) - 0.75M(\gamma)W(\gamma) + 0.36, & W(0) = 2, \\
M'(\gamma) = 0.75M(\gamma)W(\gamma) - 0.5M(\gamma), & M(0) = 1.5, \\
R'(\gamma) = 0.21W(\gamma) + 0.5M(\gamma) - 0.45R(\gamma). & R(0) = 1.
\end{cases}
$$

In order to apply the M_hWNN-LBFGS algorithm let us reformulate the above problem into an approximation solution:

$$
\begin{cases}
\tilde{W}(\gamma, \vec{\upsilon}_w) = 2.0 + (1 - e^{-(\gamma)})\,wnn(\gamma, \vec{\upsilon}_w)|_{W \to 2.0} \\
\tilde{M}(\gamma, \vec{\upsilon}_m) = 1.5 + (1 - e^{-(\gamma)})\,wnn(\gamma, \vec{\upsilon}_m)|_{M \to 1.5} \\
\tilde{R}(\gamma, \vec{\upsilon}_r) = 1.0 + (1 - e^{-(\gamma)})\,wnn(\gamma, \vec{\upsilon}_r)|_{R \to 1.0}
\end{cases}
$$

Problem 2 In the second case we have considered another non-linear WMR model by putting the values of α=0.4, β=0.21, η=0.75, $\bar{b}$=0.96, δ=0.5, θ=0.05, a_i=0, λ_1=2, λ_2=1.5, and λ_3=1. Then Eq. (1) becomes [12]

$$
\begin{cases}
W'(\gamma) = 0.4\,R(\gamma) - 0.21W(\gamma) - 0.75M(\gamma)W(\gamma) + 0.96, & W(0) = 2, \\
M'(\gamma) = 0.75M(\gamma)W(\gamma) - 0.5M(\gamma), & M(0) = 1.5, \\
R'(\gamma) = 0.21W(\gamma) + 0.5M(\gamma) - 0.45R(\gamma). & R(0) = 1.
\end{cases}
$$

Accordingly, the neural approximation solution is written as

$$
\begin{cases}
\tilde{W}(\gamma, \vec{\upsilon}_w) = 2.0 + (1 - e^{-(\gamma)})\,wnn(\gamma, \vec{\upsilon}_w)|_{W \to 2.0} \\
\tilde{M}(\gamma, \vec{\upsilon}_m) = 1.5 + (1 - e^{-(\gamma)})\,wnn(\gamma, \vec{\upsilon}_m)|_{M \to 1.5} \\
\tilde{R}(\gamma, \vec{\upsilon}_r) = 1.0 + (1 - e^{-(\gamma)})\,wnn(\gamma, \vec{\upsilon}_r)|_{R \to 1.0}
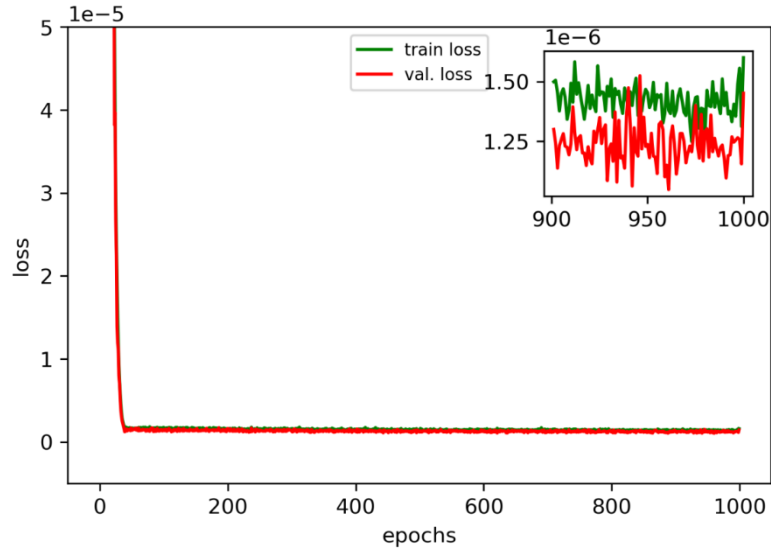\end{cases}
$$

Figure 3. Learning curves showing the result of training loss vs validation loss during the training process of M_hWNN-LBFGS with respect to epochs (Problem 1)
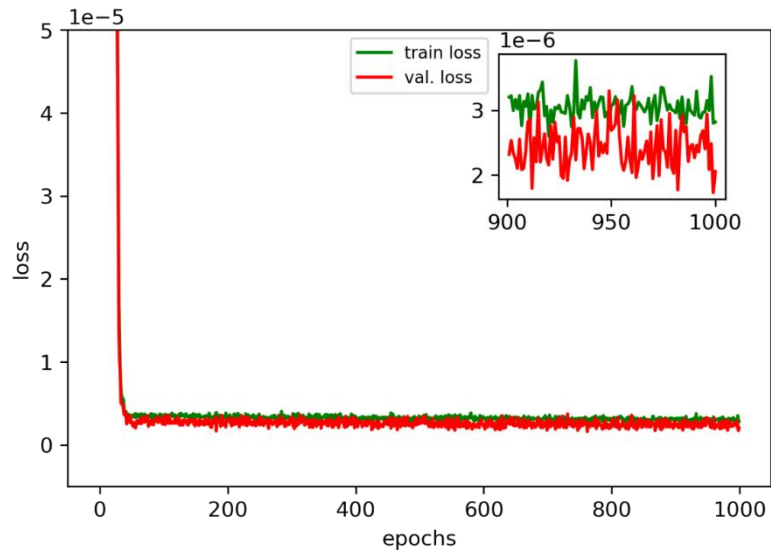

Figure 4. Learning curves showing the result of training loss vs validation loss during training process of M_hWNN-LBFGS with respect to epochs (Problem 2)

A multi-layer neural network has been constructed with a single input, single output, and three hidden layers, such that each hidden layers are set to contain 16 neurons. In addition, L-BFGS optimizer has been used to update the parameters in these test problems with a learning rate 0.01. Then the network has been trained for 100 equidistant points from t=0 sec to t=2 sec. During the training of model, we track the learning performance of each epoch through training loss and validating loss. The training and validation loss for the aforementioned two problems for 1000 epochs (last 100 epochs are depicted in subfigures) are presented in Figures 3 and 4. In other words, the validation loss in these figures represents the evolution of the network's capability for solving the WMR model. These are the values of the training and validation data set output by the loss function Eq. (14) for each epoch. Training and validation loss both continue to decrease throughout the learning process, showing the robustness of the model.

In order to, show the effectiveness of the neural algorithm, it is vital to compare the neural results obtained by the proposed algorithms with the existing results at different testing points. Moreover, to compare the proposed M_hWNN-LBFGS algorithm with the conventional methods we also run RK4 algorithms and obtained the results at 20 different testing points. The forecasted data are graphically portrayed in the plots that illustrate the reliability and consistency of M_hWNN-LBFGS. Experimental studies were reported in two subsections. First, the results of

the proposed M_hWNN-LBFGS algorithms were compared with classical results. Then statistical measures were reported for convergence analysis. Figures 5 and 6 compare the neural solutions obtained by using the proposed algorithm with the existing numerical solutions of WMR system for both problems. From the figures, it can be observed that the neural results are exactly matching with the numerical results. Tables 1 and 2 present the neural results of WMR models for all two problems at testing points $\gamma \in [0,2], \Delta\gamma = 0.1$.

In order to find the precision level, the absolute error (AE) values are delineated in Figures 7 and 8. From the box plots, we can observe that AE of Waste, Marine and Recycle lies in the range 3.8E-05 to 2.8E-04, 3.2E-05 to 5.5E-04 and 3.6E-05 to 4.4E-04 for Problem 1 and 3.4E-05 to 3.5E-04, 4.5E-06 to 4.2E-04 and 2.7E-05 to 2.6E-04 for Problem 2 respectively. One may decipher that the mode of AE values for $W(\gamma)$, $M(\gamma)$, and $R(\gamma)$ classes lie in neighbourhoods of 2E-04 and 1E-04 for Problem-1 and Problem-2 respectively. These AE values demonstrate the efficacy of the designed algorithm M_hWNN-LBFGS for solving the proposed system.
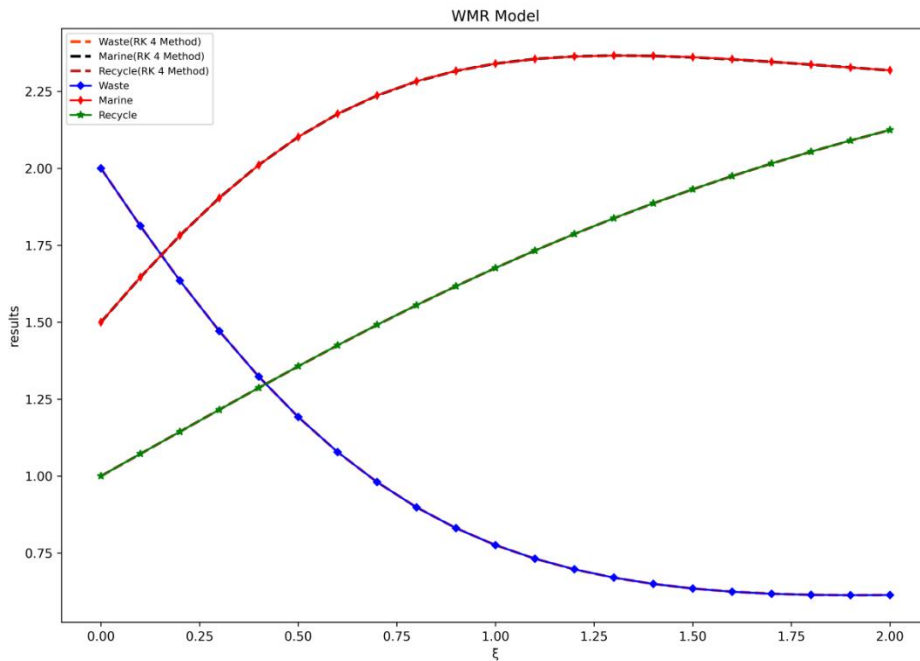

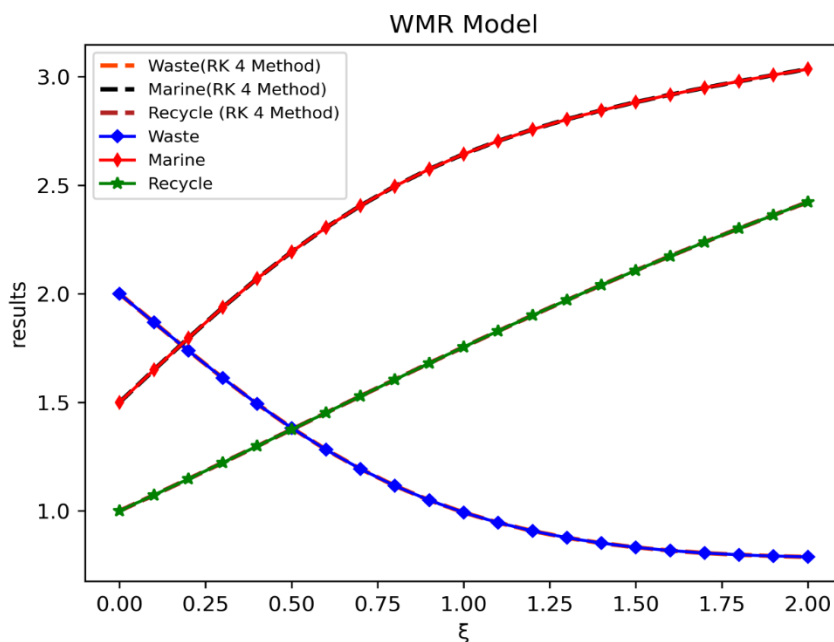**Figure 5.** Plot of RK4 solution and M_hWNN-LBFGS solution of WMR Model (Problem 1)

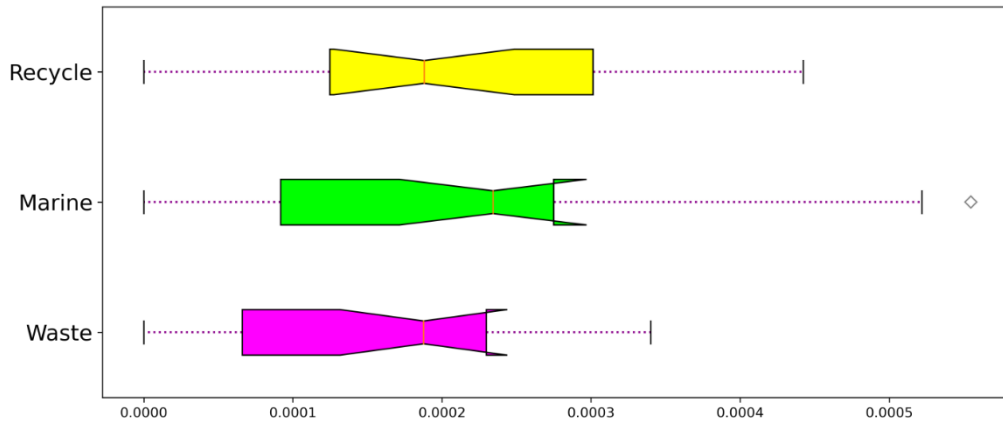**Figure 6.** Plot of RK4 solution and M_hWNN-LBFGS solution of WMR Model (Problem 2)

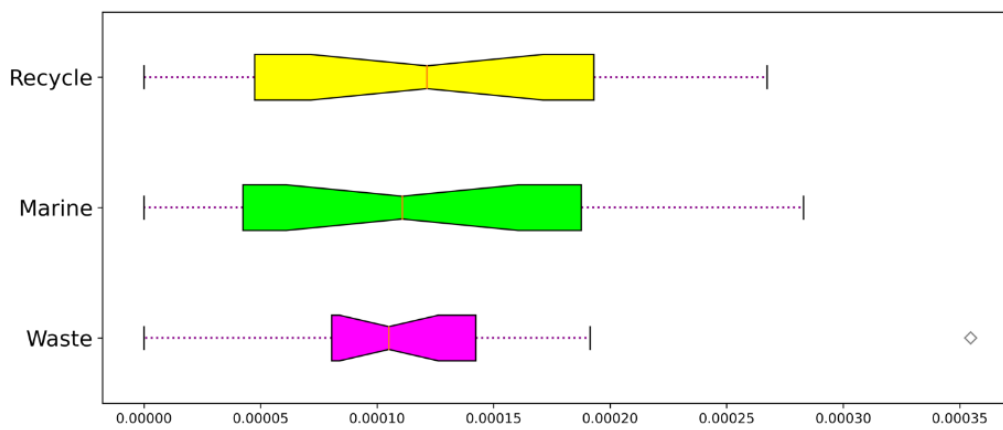**Figure 7.** Box plot of Absolute error between M_hWNN-LBFGS solution and RK4 solution (Problem 1)

Figure 8. Box plot of Absolute error between M_hWNN-LBFGS solution and RK4 solution (Problem 2)
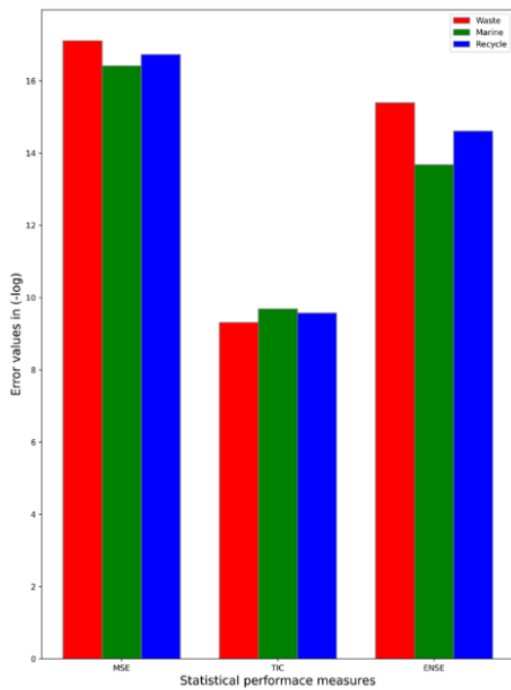

Figure 9. Performance indices based on statistical measures MSE, TIC and RMSE to solve the WMR model (Problem 1)
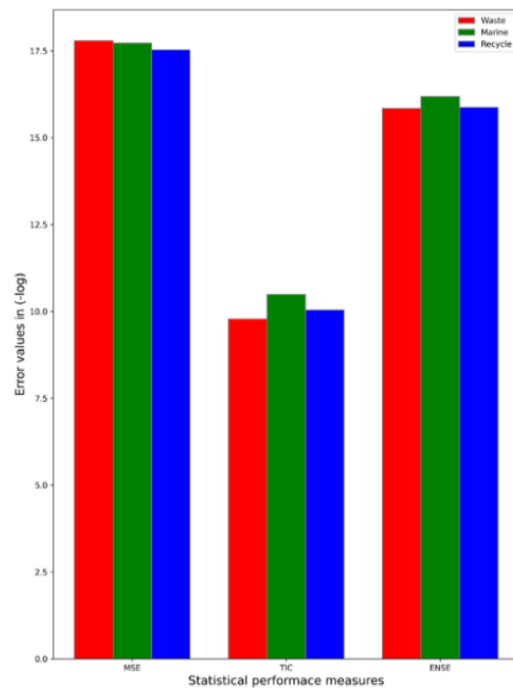

Figure 10. Performance indices based on statistical measures MSE, TIC and RMSE to solve the WMR model (Problem 2)

The outcomes of the performances based on statistical measures in terms of TIC, ENSE, and MSE are plotted in Figures 9 and 10 for the WMR non-linear system. The bar graph illustrations are used to visualize the trend of the errors. For a clear vision of values, we have presented the errors in (-log) form. It may be noted that the TIC operator magnitude for $W(\gamma)$, $M(\gamma)$, and $R(\gamma)$ lies in 6E-05 to 9E-05 and 2E-05 to 5E-05 for Problem 1 and Problem 2. Similarly, the ENSE value lies in 1E-06 to 9E-08 and the mean performance of RMSE lies in the neighbourhood of 1E-04. Moreover, the comparison of operators MAE and MAPE is shown in Tables 3 and 4. From tables, it is clearly seen that the obtained MAE errors lie in the close vicinity of 1E-04 for both cases. Whereas the MAPE varies from 1E-05 to 1E-04.

So one can evidently observe that all these global statistical measures (MAPE, TIC, RMSE, and ENSE) are close to 0, so it indicates the correctness, precision and efficacy of the M_hWNN-LBFGS model.

It is well known that after training the neural model, it can be utilized as a black box to obtain numerical results for any arbitrary points in the given domain. In this experiment, we have considered three hidden layers with 16 neurons in each hidden layer for modeling the network. One may consider more hidden layers to construct a neural model. However, by increasing the number of hidden layers and by training a network for a long time, it loses its capacity to generalize.

Table 1. M_hWNN-LBFGS solution for WMR model at testing points for $\gamma \in [0,2]$, $\Delta\gamma = 0.1$ (Problem 1)

Testing points (γ)	Waste $W(\gamma)$	Marine $M(\gamma)$	Recycle $R(\gamma)$
0.0	2.000000	1.500000	1.000000
0.1	1.812748	1.646334	1.072439
0.2	1.635189	1.781877	1.144149
0.3	1.471149	1.903975	1.215662
0.4	1.323093	2.010936	1.286773
0.5	1.192037	2.101978	1.356870
0.6	1.077988	2.177128	1.425292
0.7	0.980389	2.237094	1.491580
0.8	0.898324	2.283139	1.555530
0.9	0.830553	2.316925	1.617091
1.0	0.775548	2.340338	1.676230
1.1	0.731594	2.355293	1.732853
1.2	0.696961	2.363548	1.786828
1.3	0.670068	2.366582	1.838041
1.4	0.649599	2.365555	1.886455
1.5	0.634526	2.361362	1.932129
1.6	0.624049	2.354745	1.975207
1.7	0.617471	2.346405	2.015875
1.8	0.614074	2.337080	2.054313
1.9	0.613043	2.327554	2.090632
2.0	0.613484	2.318619	2.124825

Table 2. M_hWNN-LBFGS solution for WMR model at testing points for $\gamma \in [0,2]$, $\Delta\gamma = 0.1$ (Problem 2)

Testing points (γ)	Waste $W(\gamma)$	Marine $M(\gamma)$	Recycle $R(\gamma)$
0.0	2.000000	1.500000	1.000000
0.1	1.868125	1.649759	1.072964
0.2	1.737978	1.796453	1.146895
0.3	1.611849	1.937415	1.222118
0.4	1.492405	2.070328	1.298350
0.5	1.381929	2.193418	1.375096
0.6	1.281914	2.305565	1.451913
0.7	1.193058	2.406309	1.528507
0.8	1.115460	2.495781	1.604685
0.9	1.048818	2.574578	1.680259
1.0	0.992523	2.643625	1.754987
1.1	0.945709	2.704036	1.828586
1.2	0.907320	2.757009	1.900787
1.3	0.876230	2.803739	1.971407
1.4	0.851378	2.845357	2.040378
1.5	0.831843	2.882893	2.107744
1.6	0.816849	2.917246	2.173612
1.7	0.805692	2.949172	2.238081
1.8	0.797651	2.979268	2.301173
1.9	0.791944	3.007966	2.362765
2.0	0.787732	3.035539	2.422549

Table 3. Error values of waste, marine and recycle at testing points for (Problem-1)

MAE	1.6562380952381668 E-04	2.2793714285719777 E-04	1.9516619047618304 E-04
MAPE	2.0698206226719160 E-04	1.0256963610789712 E-04	1.2201287228477674 E-04

Table 4. Error values of waste, marine and recycle at testing points for (Problem-2)

MAE	1.1274571428569284 E-04	1.1430809523808409 E-04	1.2962142857148295 E-04
MAPE	1.0470961564736848 E-04	4.3931601159521410 E-05	7.8345688953517170 E-05

6. Conclusion

The present work shows the application of neural techniques for simulating waste plastic management of ocean in order to conserve marine ecosystems. The advantages of this proposed method are examined by solving different cases which describe various phenomena in oceanography. The values of statistical measures that are very close to 0, confirms the reliability and correctness of M_hWNN-LBFGS. The excellent agreement between the neural results and traditional numerical methods shows that the newly developed M_hWNN-LBFGS algorithm is extremely accurate for simulating the non-linear WMR model.

The ML algorithm addressed in this article is generic and can be useful for solving relevant problems emerging in various other engineering applications such as, epidemic model [31], fuzzy space-fractional tele-graph model [32], fractional order coupled wave equations [33], astrophysics model [34], etc.

Author Contributions

Conceptualization, A.K Sahoo; methodology, A.K Sahoo and S. Chakraverty; validation, A.K Sahoo and S. Chakraverty; formal analysis, A.K Sahoo; investigation, A.K Sahoo and S. Chakraverty; writing—original draft preparation, A.K Sahoo; writing—review and editing, S. Chakraverty; supervision, S. Chakraverty.

All authors have read and agreed to the published version of the manuscript.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgements

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] N. Portillo Juan and V. Negro Valdecantos, "Review of the application of Artificial Neural Networks in ocean engineering," *Ocean Eng.*, vol. 259, Article ID: 111947, 2022. https://doi.org/10.1016/J.OCEANENG.2022.111947.

[2] "The ocean plastic pollution challenge: Towards solutions in the UK Headlines," Grantham, 2016, https://www.imperial.ac.uk/grantham/publications/.

[3] G. Bishop, D. Styles, and P. N. L. Lens, "Recycling of European plastic is a pathway for plastic debris in the ocean," *Environ Int.*, vol. 142, Article ID: 105893, 2020. https://doi.org/10.1016/J.ENVINT.2020.105893.

[4] S. Chakraverty and S. K. Jeswal, *Applied Artificial Neural Network Methods for Engineers and Scientists*, India: World Scientific, 2021.

[5] A. K. Sahoo and S. Chakraverty, "Multilayer unsupervised symplectic artificial neural network model for solving Duffing and Van der Pol–Duffing oscillator equations arising in engineering problems," *Modeling Comput. Vibration Problems*, vol. 2, pp. 1-13, 2021. https://doi.org/10.1088/978-0-7503-3487-7ch3.

[6] L. S. Tan, Z. Zainuddin, and P. Ong, "Wavelet neural networks based solutions for elliptic partial differential equations with improved butterfly optimization algorithm training," *Appl. Soft Comput.*, vol. 95, Article ID:

106518, 2020. https://doi.org/10.1016/j.asoc.2020.106518.

[7] H. Badem, A. Basturk, A. Caliskan, and M. E. Yuksel, "A new hybrid optimization method combining artificial bee colony and limited-memory BFGS algorithms for efficient numerical optimization," *Appl. Soft Comput.*, vol. 70, pp. 826-844, 2018. https://doi.org/10.1016/J.ASOC.2018.06.010.

[8] W. S. Mcculloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, pp. 115-133, 1943. https://doi.org/10.1007/BF02478259.

[9] M. J. Alizadeh and M. R. Kavianpour, "Development of wavelet-ANN models to predict water quality parameters in Hilo Bay, Pacific Ocean," *Mar Pollut Bull.*, vol. 98, no. 1-2, pp. 171-178, 2015. https://doi.org/10.1016/j.marpolbul.2015.06.052.

[10] C. Gu, J. Qi, Y. Zhao, W. Yin, and S. Zhu, "Estimation of the mixed layer depth in the indian ocean from surface parameters: A clustering-neural network method," *Sensors*, vol. 22, no. 15, Article ID: 5600, 2022. https://doi.org/10.3390/s22155600.

[11] B. Xue, B. Huang, W. Wei, G. Chen, H. Li, N. Zhao, and H. Zhang, "An efficient deep-sea debris detection method using deep neural networks," *IEEE J. Sel. Top Appl Earth Obs. Remote Sens.*, vol. 14, pp. 12348-12360, 2021. https://doi.org/10.1109/JSTARS.2021.3130238.

[12] M. Al Nuwairan, Z. Sabir, M. A. Z. Raja, and A. Aldhafeeri, "An advance artificial neural network scheme to examine the waste plastic management in the ocean," *AIP Adv.*, vol. 12, no. 4, Article ID: 045211, 2022. https://doi.org/10.1063/5.0085737.

[13] H. Lee and I. S. Kang, "Neural algorithm for solving differential equations," *J. Comput. Phys.*, vol. 91, no. 1, pp. 110-131, 1990. https://doi.org/10.1016/0021-9991(90)90007-N.

[14] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 987-1000, 1998. https://doi.org/10.1109/72.712178.

[15] Y. Wen, T. Chaolu, and X. Wang, "Solving the initial value problem of ordinary differential equations by Lie group based neural network method," *PLoS One*, vol. 17, no. 4, Article ID: e0265992, 2022. https://doi.org/10.1371/journal.pone.0265992.

[16] A. Verma and M. Kumar, "Numerical solution of Bagley–Torvik equations using Legendre artificial neural network method," *Evol. Intell.*, vol. 14, no. 4, pp. 2027-2037, 2021. https://doi.org/10.1007/s12065-020-00481-x.

[17] D. Mohapatra and S. Chakraverty, "Initial value problems in Type-2 fuzzy environment," *Math Comput Simul*, vol. 204, pp. 230-242, 2023. https://doi.org/10.1016/j.matcom.2022.08.002.

[18] H. Liu, B. Xing, Z. Wang, and L. Li, "Legendre neural network method for several classes of singularly perturbed differential equations based on mapping and piecewise optimization technology," *Neural Process Lett.*, vol. 51, no. 3, pp. 2891-2913, 2020. https://doi.org/10.1007/s11063-020-10232-9.

[19] A. K. Sahoo and S. Chakraverty, *Studies in Computational Intelligence*, Singapore: Springer, 2022. https://doi.org/10.1007/978-981-16-4713-0_6.

[20] I. Ahmad, M. A. Z. Raja, M. Bilal, and F. Ashraf, "Neural network methods to solve the Lane–Emden type equations arising in thermodynamic studies of the spherical gas cloud model," *Neural Comput. Appl.*, vol. 28, pp. 929-944, 2017. https://doi.org/10.1007/s00521-016-2400-y.

[21] Q. Liu, X. Yu, and Q. Feng, "Fault diagnosis using wavelet neural networks," *Neural Process. Lett.*, vol. 18, no. 2, pp. 115-123, 2003. https://doi.org/10.1023/A:1026231814341.

[22] H. Abghari, H. Ahmadi, S. Besharat, and V. Rezaverdinejad, "Prediction of daily pan evaporation using wavelet neural networks," *Water Resour. Manag.*, vol. 26, no. 12, pp. 3639-3652, 2012. https://doi.org/10.1007/s11269-012-0096-z.

[23] N. M. Pindoriya, S. N. Singh, and S. K. Singh, "An adaptive wavelet neural network-based energy price forecasting in electricity markets," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1423-1432, 2008. https://doi.org/10.1109/TPWRS.2008.922251.

[24] V. T. Yen, W. Y. Nan, and P. van Cuong, "Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 6945-6958, 2019. https://doi.org/10.1007/s00521-018-3520-3.

[25] M. Malekzadeh, J. Sadati, and M. Alizadeh, "Adaptive PID controller design for wing rock suppression using self-recurrent wavelet neural network identifier," *Evol. Syst.*, vol. 7, no. 4, pp. 267-275, 2016. https://doi.org/10.1007/s12530-015-9143-3.

[26] Z. Sabir, M. A. Z. Raja, J. L. G. Guirao, and M. Shoaib, "A novel design of fractional Meyer wavelet neural networks with application to the nonlinear singular fractional Lane-Emden systems," *Alex. Eng. J.*, vol. 60, no. 2, pp. 2641-2659, 2021. https://doi.org/10.1016/j.aej.2021.01.004.

[27] M. Wu, J. Zhang, Z. Huang, X. Li, and Y. Dong, "Numerical solutions of wavelet neural networks for fractional differential equations," *Math Methods Appl. Sci.*, vol. 46, no. 3, pp. 3031-3044, 2023. https://doi.org/10.1002/mma.7449.

[28] D. Veitch, "Wavelet Neural Networks and their application in the study of dynamical systems," Master

Dissertation, York University, UK, 2005.

[29] J. Zhang, G. G. Walter, Y. Miao, and W. N. Wayne Lee, "Wavelet neural networks for function learning," *IEEE Trans. Signal Process.*, vol. 43, no. 6, pp. 1485-1497, 1995. https://doi.org/10.1109/78.388860.

[30] A. K. Sahoo and S. Chakraverty, "Machine intelligence in dynamical systems: A state of art review," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 12, no. 4, Article ID: e1461, 2022. https://doi.org/10.1002/widm.1461.

[31] W. Weera, T. Botmart, T. La-inchua, Z. Sabir, R. A. S. Núñez, M. Abukhaled, and J. L. G. Guirao, "A stochastic computational scheme for the computer epidemic virus with delay effects," *AIMS Math.*, vol. 8, no. 1, pp. 148-163, 2023. https://doi.org/10.3934/math.2023007.

[32] S. Tapaswini and D. Behera, "Analysis of imprecisely defined fuzzy space-fractional telegraph equations," *Pramana*, vol. 94, no. 1, Article ID: 32, 2020. https://doi.org/10.1007/s12043-019-1889-x.

[33] S. Dubey and S. Chakraverty, "Application of modified extended tanh method in solving fractional order coupled wave equations," *Math Comput. Simul.*, vol. 198, pp. 509-520, 2022. https://doi.org/10.1016/j.matcom.2022.03.007.

[34] A. Verma and M. Kumar, "Numerical solution of third-order Emden–Fowler type equations using artificial neural network technique," *Eur. Phys J. Plus*, vol. 135, no. 9, Article ID: 751, 2020. https://doi.org/10.1140/epjp/s13360-020-00780-3.