Results in Applied Mathematics 22 (2024) 100457

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics

# Numerical solutions of sea turtle population dynamics model by using restarting strategy of PINN-Adam

Danang A. Pratama [a], Maharani A. Bakar [a,b,*], Ummu Atiqah Mohd Roslan [b,d], Sugiyarto Surono [c], A. Salhi [e]

[a] Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, Kuala Nerus, Terengganu, 21030, Malaysia
[b] Special Interest Group Modelling and Data Analytics, Universiti Malaysia Terengganu, Kuala Nerus, Terengganu, 21030, Malaysia
[c] Mathematic Department, Faculty of Applied Sciences and Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[d] Institute of Oceanography and Environment, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia
[e] Department of Mathematical Sciences, University of Essex, Wivenhoe Park, Colchester, CO43SQ, UK

ARTICLE INFO

*Keywords:*
Lotka–Volterra
Predator–prey
Sea turtles
Population dynamics
PINN
Adam optimizer
Restarting strategy

ABSTRACT

The Lotka–Volterra predator–prey system for dynamic sea turtle population is solved using r-PINN-Adam method, a novel approach which combines Physics-Informed Neural Network (PINN) with restarting strategy. This method allows us to monitor the loss function values of PINN such that when there is no progress made, we stop the process and pick a good value to be used in the next process. Subsequently, the training time decreases and the accuracy increases. The numerical solutions are compared to the popular Runge–Kutta method in terms of correctness which presented graphically. Simulation results also displayed in terms of trainable parameters and optimal loss function performance. The research highlights the robustness and superiority of the proposed method, positioning it as a valuable tool for sea turtle conservation efforts.

## 1. Introduction

Sea turtles are a group of reptiles that have adapted to life in the ocean. There are seven species of sea turtles, all of which are classified as endangered or critically endangered. They can be found in all of the world's oceans, from the warm tropical waters of the Caribbean to the cold waters of the Arctic. Sea turtles are known for their distinctive shells, which are composed of two parts: the carapace (upper shell) and the plastron (lower shell). The shell provides protection from predators and helps regulate the turtle's body temperature. Sea turtles are also known for their long migrations, which can take them thousands of miles across the ocean. They use the Earth's magnetic field to navigate and can return to the same beach where they were born to lay their eggs.

Sea turtles face several natural predators during their life cycle [1,2]. The specific predators can vary depending on the species and the location, but some common ones include:

1. Predatory birds such as seagulls and frigate birds that prey on hatchlings as they emerge from their nests on the beach.
2. Crabs and other small predators that can dig into nests and consume eggs.
3. Sharks, particularly tiger sharks and great white sharks, are known to attack adult sea turtles. They often target the turtles as they swim near the surface of the water.
4. Saltwater crocodiles in some areas can also prey on sea turtles.

* Corresponding author at: Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, Kuala Nerus, Terengganu, 21030, Malaysia.
*E-mail address:* maharani@umt.edu.my (M.A. Bakar).

https://doi.org/10.1016/j.rinam.2024.100457
Received 17 February 2024; Received in revised form 11 April 2024; Accepted 12 April 2024
Available online 25 April 2024

5. In some places, large fish such as groupers and barracudas can also prey on sea turtles.
6. In addition, some species of sea turtles, such as the green sea turtle, are also preyed upon by large mammals such as jaguars and coyotes when they come to shore to lay their eggs.

Sea turtles have a complex life cycle that involves distinct stages of development. The stage structure of sea turtles can be divided into two main stages: the pelagic stage and the benthic stage. During the pelagic stage, sea turtle hatchlings emerge from their nests on the beach and enter the open ocean. They spend several years swimming in the ocean, often traveling thousands of miles and facing many hazards such as predators and ocean currents. As they grow and mature, sea turtles undergo a metamorphosis and transition to the benthic stage. This is when they return to coastal waters and habitats such as seagrass beds and coral reefs. During this stage, sea turtles continue to grow and mature, and eventually reach sexual maturity. Within the benthic stage, there are also different life stages that sea turtles go through. For example, juvenile sea turtles are different from adults in terms of their habitat and feeding preferences. Green sea turtles, for example, undergo an herbivorous shift as they grow and become adults, while hawksbill turtles remain carnivorous throughout their life. The stage structure of sea turtles is important to consider in conservation efforts, as threats and impacts to sea turtles can vary depending on the life stage. For example, hatchlings are vulnerable to predation on nesting beaches, while adult turtles face threats such as bycatch in fishing gear and habitat destruction. Understanding the different life stages of sea turtles is crucial for developing effective conservation strategies to protect them throughout their life cycle [3,4].

The mathematical model of the prey–predator system provides a useful tool for studying the population dynamics of sea turtles. Its solutions can provide valuable insights into the mechanisms that govern the growth and decline of sea turtle populations. The interaction between predators and prey in any environment was first introduced by Lotka and Volterra in 1926 [5]. The Lotka–Volterra equations can describe how the populations of predators and sea turtles change over time, based on a set of parameters such as the growth rate of the sea turtles, the efficiency of the predator in catching sea turtles, and the death rates of both predators and sea turtles. Several researchers have proposed a prey–predator system that incorporates the stage-structured life of sea turtles, recognizing the significance of studying their various life phases. Khairuddin et al. [6], for instance, included the eggs laid in a nest and the hatchlings or baby sea turtles in their model, and also accounted for natural predation by marine predators. On the other hand, Wei et al. [7] considered both the egg and adult stages, and also determined the sex of the embryo.

The dynamics of predator–prey systems can be quite complex, and analytical solutions are often difficult or even impossible to obtain either because of not yet able to find or because it cannot be expressed by means of elementary functions [8,9]. In such cases, numerical methods provide an effective alternative for solving these systems. One common approach is to use classical numerical methods, such as the Euler or Runge–Kutta methods, to approximate the solutions to the differential equations governing the system. The Euler method is used in [10] to explore the use of stochastic processes in the extinction behavior of the interacting populations of the Lotka–Volterra model. Paul et al. [11] used the Runge–Kutta–Fehlberg method (RKF) in predator–prey systems with complex dynamic characteristics and compared it with the Laplace Adomian Decomposition method (LADM), resulting in the finding that RKF is a more accurate and reliable numerical technique than LADM in the models. In another approach, Du et al. [12] used the sine function interpolation collocation method for solving a class of predator–prey systems with complex dynamic characteristics. However, classical methods often rely on the discretization of the domain, which can be computationally expensive, especially when dealing with high-dimensional problems or problems with complex geometries [13,14]. Furthermore, these classical methods are not suitable for real-time applications and can be time-consuming to adapt to new applications, particularly for those who are not experts in numerical mathematics and computational modeling [15].

Recently, Artificial Neural Networks (ANNs) have emerged as an alternative approach to problem-solving, demonstrating remarkable success in various fields, including differential equations (DEs). ANNs have been utilized to solve both ordinary (ODEs) and partial differential equations (PDEs), taking into account trial solutions that satisfy the initial and boundary conditions [16]. Furthermore, ANNs can be integrated with any optimization method to enhance the results. For instance, Malek and Beidokhti [17] combined ANNs with the Nelder–Mead simplex method to find solutions for high-order PDEs. Lastly, an investigation of a three-species food chain via the Lotka–Volterra model using Gudermannian neural networks (GNNs) combined with a Genetic Algorithm and Active-Set Approach (GA-ASA) was conducted and compared with the Runge–Kutta method [18]. The results showed good performance across different performance metrics.

In this paper, we investigate numerical solutions for the population dynamics of sea turtles using a method known as the Physics-Informed Neural Network (PINN) [19]. PINN is an ANN-based method that incorporates the physical laws of the model into the loss function as a regularization term, thereby enhancing the performance of the ANN. Currently, PINN is attracting great attention from researchers and is being applied in various fields of research. Several modifications to the original PINN method have been proposed to enhance its performance. Jagtap et al. introduced both globally [20] and locally [21] adaptive activation functions in PINN. These functions have demonstrated better learning capabilities compared to the traditional fixed activation function. McClenny and Braga-Neto modified the loss function by adding adaptive weight, resulting in the self-adaptive PINN [22]. This method was further improved by Pratama et al. who used a hybrid of the Genetic Algorithm (GA) and the L-BFGS method as an optimizer instead of Adam. This resulted in improved accuracy for nine different Partial Differential Equations (PDE) [23].

Moreover, we have modified PINN by introducing a restarting process. This approach monitors the loss value during the training process and automatically stops the iteration if there is no further improvement in the loss function. It then restarts the iteration from the beginning, using the best trainable parameters from the previous cycle in the next cycle. We combined this approach with the Adam optimizer to form r-PINN-Adam. The effectiveness of this technique has been demonstrated by applying it to solve the PDE for thermal analysis, studying the temperature behavior surrounding breast cancer [24]. This technique can increase the model's accuracy and significantly reduce wasted time compared to the original PINN, which employs a fixed iteration. The salient features of the paper are summarized as follows:

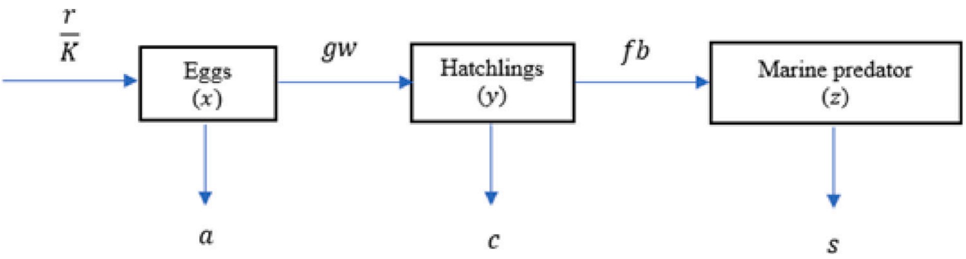



**Fig. 1.** The flow diagram for the relationship between eggs, hatchlings, and marine predator.

1. A mathematical model for the population dynamics of sea turtles has been formulated and analyzed. This model studies the influence of variations in the rate of eggs laid by a mother turtle ($r$), the conversion rate from eggs to hatchlings ($g$), the predation rate by marine predators ($b$), and the successful rate of eggs that hatch ($w$).
2. A novel r-PINN is used to construct a model of approximate solutions for the population dynamics system of the sea turtle model. A loss function, based on mean square errors (MSE), is designed and monitored to update the trainable parameters (weights and biases) with the help of the optimization mechanism of the Adam optimizer. Additionally, to evaluate the performance of our proposed algorithm, we have also designed another loss function based on mean absolute errors (MAE) and root mean squared errors (RMSE).
3. The numerical results obtained from the proposed r-PINN-Adam method have been compared with the fourth-order of Runge–Kutta method (RK4) to validate the accuracy of the designed approach.
4. We aim to design several proposed actions related to our findings and the impact of variations in the parameters $r$, $g$, $b$, and $w$.
5. The approximate solutions, trainable parameters, computational time, and performance of the loss function obtained by the r-PINN-Adam algorithm for the population dynamics system of sea turtles are demonstrated through various graphs and tables. These visualizations highlight the dominance and robustness of the proposed method in solving real-world problems.

Overall, this study is organized as follows. Section 2 presents the mathematical modeling of the population dynamics of sea turtles, using the Lotka–Volterra equation. Section 3 outlines the numerical implementation of the r-PINN-Adam approach for solving the population dynamics of sea turtles. Section 4 discusses the results of variations in the rate of eggs laid by a mother turtle ($r$), the conversion rate from eggs to hatchlings ($g$), the predation rate by marine predators ($b$), and the successful rate of eggs that hatch ($w$). Finally, Section 5 concludes by summarizing the findings.

## 2. Derivation of mathematical model of population dynamics

In this research, we consider a mathematical model proposed by Khairuddin et al. [6]. To study the dynamics of the sea turtle population, a prey–predator model has been constructed with three compartments: (1) the number of eggs laid in the nest (denoted by $x$), (2) the number of hatchlings (denoted by $y$), and (3) the number of marine predators (denoted by $z$). Before constructing the model, the following assumptions were considered:

1. The population of sea turtle eggs increases by its natural growth rate $r$. It is assumed that the number of eggs evolved follows the logistic growth, i.e., $rx(1-\frac{x}{K})$ since this is more biologically reasonable compared to the exponential growth curve. These numbers are limited by the maximum carrying capacity $K$.
2. It is assumed that the number of eggs may decrease due to a hatch failure rate ($a$). The eggs that successfully hatch were then transformed into hatchlings at a rate $w$.
3. The hatchling population ($y$) grows as a result of eggs successfully hatching. The transition from eggs to hatchlings involves an energy conversion process with a rate $g$.
4. The number of hatchlings may decrease due to the predation rate $b$. On the other hand, the hatchlings that managed to escape from the predator are converted into adult stage with rate $c$.
5. The marine predator increases due to its consumption on the hatchlings. This is because by consuming the hatchlings, energy will be transferred from the hatchlings to the predator (with rate $f$). Additionally, the marine predator population may decrease due to its natural death rate $s$.

Therefore, with the above assumptions, the relationship between eggs, hatchlings, and the marine predator can be sketched as in Fig. 1.

**Table 1**
The definition of all parameters in model Eq. (1).

| Parameters | Description | Values | References |
|---|---|---|---|
| $r$ | Rate of eggs laid by a mother turtle | 1.3 | [25] |
| $K$ | Carrying capacity for eggs in the nest | 700 | [25] |
| $a$ | Failing rate of eggs to hatch | 0.2 | [26] |
| $w$ | Successful rate of eggs that hatched | 0.53 | [27] |
| $c$ | Survival rate of hatchlingsthat managed to escape from predator | 0.001 | [28] |
| $b$ | Predation rate for hatchlings by marine predator | 0.015 | [29] |
| $g$ | Conversion rate from eggs to hatchlings | 1 | [29] |
| $s$ | Natural death rate for marine predator | 0.7 | Assumed |
| $f$ | Energy conversion rate from hatchlingsto marine predator | 1 | Assumed |

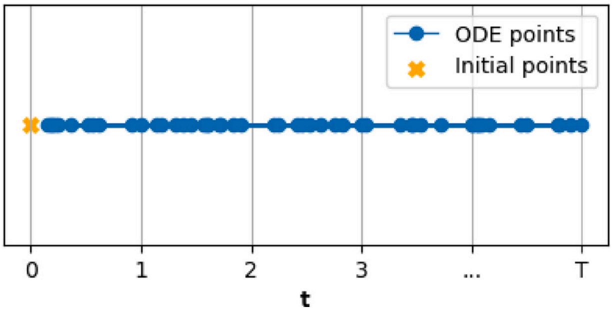



**Fig. 2.** Initial and ODE points in system ODE population dynamic model.

From Fig. 1, a system of ordinary differential equations can be formulated as follows:

$$\begin{aligned}\frac{dx}{dt} &= rx\left(1-\frac{x}{K}\right) - ax - wx,\\ \frac{dy}{dt} &= gwx - cy - byz,\\ \frac{dz}{dt} &= fbyz - sz.\end{aligned} \tag{1}$$

The description of all parameters is provided in Table 1. The initial conditions are $x(0) \leq 0, y(0) \leq 0$, and $z(0) \leq 0$. For the system to be biologically meaningful, the solutions must always be positive and bounded. We compared our solution with the solutions obtained by fourth-order Runge–Kutta method (RK4) to validate the correctness and effectiveness of our proposed method.

## 3. Restarting PINN-Adam for solving population dynamic model

We propose a solution to the prey–predator model defined in Eq. (1), based on the restarting PINN-Adam (r-PINN-Adam) method, which modifies PINN-Adam with a restarting strategy. This method allows us to find better solutions in a shorter time compared to the original PINN. Technically, we run the PINN-Adam algorithm for a certain number of iterations, then monitor the loss value for a few more iterations using earlystopping by Tensorflow. If there is no improvement made during the monitoring, we stop and select the weights that result in a smaller value of the loss function. These new weights are used for the next cycle. The process is repeated until it meets the maximum number of cycles. The details of this process are explained in the next section. The procedure of r-PINN-Adam to solve the model of the dynamical system is also explained in the next section.

### *3.1. Define the ODE and initial points*

The ODE system of population dynamics is based on Eq. (1). To solve it numerically, we divide the domain into some points, $t_i$, $i = 0, 1, \dots, n$, as illustrated in Fig. 2. These points are later divided into training and testing points for the Artificial Neural Network (ANN) input. The number of training points is usually between 60% and 80% of the total available points.

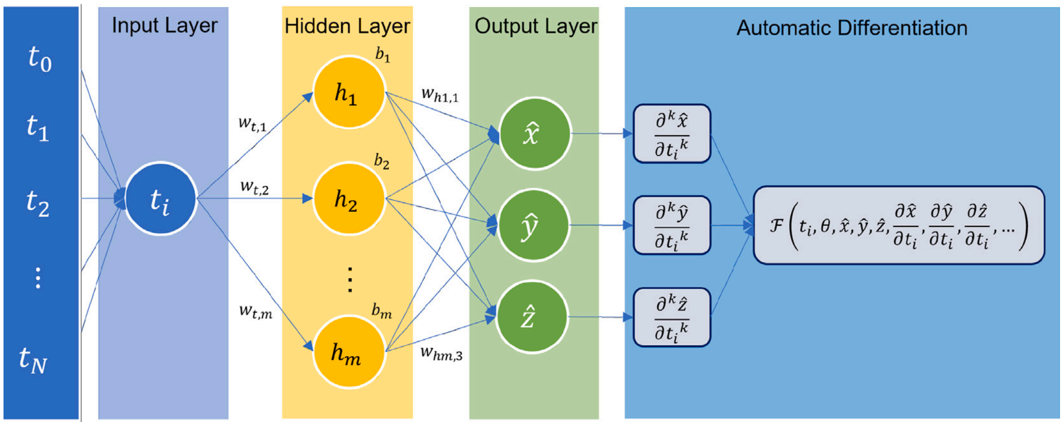



**Fig. 3.** ANN with single input $(t_i)$, one hidden layer with $m$ neurons, and three outputs $\hat{x}(t_i;\theta)$, $\hat{y}(t_i;\theta)$ and $\hat{z}(t_i;\theta)$.

### 3.2. Define the ANN architecture

As an illustration, suppose we have a three-layered perceptron that takes $(t_i)$ as input, has one hidden layer with $m$ neurons, and produces three outputs $\hat{x}(t_i;\theta)$, $\hat{y}(t_i;\theta)$ and $\hat{z}(t_i;\theta)$, which serve as an approximate system for the ODE function. Fig. 3 visualizes this architecture.

To obtain the numerical solutions of $x(t_i)$, $y(t_i)$ and $z(t_i)$ using the approximate functions $\hat{x}(t_i;\theta)$, $\hat{y}(t_i;\theta)$ and $\hat{z}(t_i;\theta)$, the parameter $\theta$, which consists of weights and biases must be updated to minimize the loss function. In this study, Adam optimizer will be used to update $\theta$.

### 3.3. Basic PINN in solving differential equation problem

Generally, the process of using PINN to solve a differential equation model begins by initializing the parameter $\theta$, which forms a linear combination with the inputs $t_i$, where $i = 0, \ldots, N$. In this case, we employ a random uniform distribution to initiate the $\theta$ from the hidden layer to the output layer. This outcome is subsequently utilized to facilitate the learning process, [30]. The inputs are then processed in the hidden layer by formula (2) provided below,

$$h_j = t_i w_{t,j} + b_j, \ with\ j = 1, \ldots, m. \tag{2}$$

where $h_j$ is the $j$th neuron from the hidden layer, $w_{t,j}$ is the weight that connects the input layer to the hidden layer, and $b_j$ is the bias. The activation function used in the hidden layer is crucial in an ANN. In this architecture, the hyperbolic tangent ($tanh$) activation function is employed, as it has been empirically shown to offer a superior approximation compared to alternative functions [31],

$$f(h_j) = \frac{e^{h_j} - e^{h_j}}{e^{h_j} + e^{h_j}}. \tag{3}$$

The next step involves obtaining $\hat{x}(t_i;\theta)$, $\hat{y}(t_i;\theta)$ and $\hat{z}(t_i;\theta)$ through the output layer using the following formula:

$$\begin{aligned} \hat{x}(t_i;\theta) &= \Sigma_{j=1}^{m} f(h_j) w_{h_j,1}, \\ \hat{y}(t_i;\theta) &= \Sigma_{j=1}^{m} f(h_j) w_{h_j,2}, \\ \hat{z}(t_i;\theta) &= \Sigma_{j=1}^{m} f(h_j) w_{h_j,3}. \end{aligned} \tag{4}$$

The $k$th derivatives of $\hat{x}(t_i;\theta)$, $\hat{y}(t_i;\theta)$ and $\hat{z}(t_i;\theta)$ can be computed using automatic differentiation (AD) as shown in Eq. (5). This computation is implemented using the open-source deep learning framework, TensorFlow [32].

$$\begin{aligned} \frac{\partial^k}{\partial t_i^k} \hat{x}(t_i;\theta) &= \Sigma_{j=1}^{m} \frac{\partial^k}{\partial t_i^k} f(h_j) w_{h_j,1}, \\ \frac{\partial^k}{\partial t_i^k} \hat{y}(t_i;\theta) &= \Sigma_{j=1}^{m} \frac{\partial^k}{\partial t_i^k} f(h_j) w_{h_j,2}, \\ \frac{\partial^k}{\partial t_i^k} \hat{z}(t_i;\theta) &= \Sigma_{j=1}^{m} \frac{\partial^k}{\partial t_i^k} f(h_j) w_{h_j,3}. \end{aligned} \tag{5}$$

The loss function of the basic PINN is expressed as follows:

$$\mathcal{L}(\theta) = MSE_{x0} + MSE_{y0} + MSE_{z0} + MSE_x + MSE_y + MSE_z, \tag{6}$$

where $MSE_{x0}$, $MSE_{y0}$, $MSE_{z0}$, $MSE_x$, $MSE_y$, and $MSE_z$ are the MSE-based loss function associated with the initial conditions, as given in Eq. (7).

$$\begin{cases} MSE_{x0} = \left[\hat{x}(t_0;\theta) - x(t_0)\right]^2, \\ MSE_{y0} = \left[\hat{y}(t_0;\theta) - y(t_0)\right]^2, \\ MSE_{z0} = \left[\hat{z}(t_0;\theta) - z(t_0)\right]^2, \\ MSE_x = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t_i}\hat{x}(t_i;\theta) - r\hat{x}(t_i;\theta)\left(1 - \frac{\hat{x}(t_i;\theta)}{K}\right) + a\hat{x}(t_i;\theta) + w\hat{x}(t_i;\theta)\right]^2, \\ MSE_y = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t_i}\hat{y}(t_i;\theta) - gw\hat{x}(t_i;\theta) + c\hat{y}(t_i;\theta) + b\hat{y}(t_i;\theta)\hat{z}(t_i;\theta)\right]^2, \\ MSE_z = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t_i}\hat{z}(t_i;\theta) - fb\hat{y}(t_i;\theta)\hat{z}(t_i;\theta) + s\hat{z}(t_i;\theta)\right]^2, \end{cases} \tag{7}$$

where $N$ is the total number of sample data points. Moreover, $t_0$ and $t_i$ are the initial and ODE points, respectively, which are randomly distributed in the domain.

The most effective approach to achieving an optimal value for the loss function involves minimizing it towards zero. This is accomplished by employing the Adam optimizer, which iteratively adjusts the $\theta$ values to converge towards the optimal solution.

### 3.4. Adam optimizer

In the next stage, the trainable parameter $\theta$ is updated in such a way that it minimizes the loss function by using the Adam optimizer. The process of updating trainable parameters using this method can be explained using the following equation proposed by [33].

$$\theta_{l+1} = \theta_l - \frac{\alpha_{l+1} m_{l+1}}{\sqrt{v_{l+1}} + \epsilon}, \tag{8}$$

where $\alpha_l$ is the adaptive learning rate, where the default initial value is $\alpha_l = 0.001$ and described as follows:

$$\alpha_{l+1} = \alpha_l \frac{\sqrt{1-\beta_l^1}}{1-\beta_l^2}, \tag{9}$$

and

$$\begin{aligned} m_{l+1} &= \beta^1 m_l + (1-\beta^1)\nabla_\theta \mathcal{L}(\theta_l), \\ v_{l+1} &= \beta^2 v_l + (1-\beta^2)[\nabla_\theta \mathcal{L}(\theta_l)]^2. \end{aligned} \tag{10}$$

Both $m_l$ and $v_l$ are initialized by 0, and other parameters are described by the following:
$l$: iteration $(0, 1, 2, \ldots)$,
$\theta_l$: vector of trainable parameters in the current iteration.
$\theta = [w^l_{t,1}, \ldots, w^l_{t,m}, b^l_1, \ldots, b^l_m, w^l_{h1,1}, \ldots, w^l_{hm,3}]$,
$\beta^1, \beta^2 \in [0, 1)$: exponential decay rates for moment estimates,
$\epsilon$: very small number to prevent divided by zero,
$\nabla_\theta \mathcal{L}(\theta_l)$: the gradient of the loss function regarding the trainable parameters.

The ANN architecture is iteratively refined by employing the Adam optimizer to generate a new $\theta_{l+1}$. The process, defined by Eqs. (2)–(10), is repeated until a termination condition is met. This termination condition could be based on reaching a specific number of iterations or satisfying a predetermined error threshold between consecutive solutions obtained.

### 3.5. Restarting strategy in algorithm PINN-Adam

Typically, when we execute the PINN-Adam algorithm to solve differential equations (DE), we employ certain stopping criteria such as the number of iterations or a specific error value. Unfortunately, these schemes cannot guarantee that the optimal solution will be achieved. Moreover, a significant amount of computational time is expended to achieve a small value of the loss function. A restarting strategy is designed to address this issue. Adopt the scheme from [34], we run PINN-Adam for a certain number of iterations, say 100 iterations, and then monitor the movement of the loss function value of the Adam optimizer by early-stopping technique by Tensorflow. If the loss value does not decrease in several subsequent iterations, the process will stop and the algorithm will automatically restart. This is what we refer to as one cycle of running. In one cycle, we can find the best $\theta$. We use these as the input for the next cycle. The cycle process stops once a certain maximum number of cycles is met. We describe the entire process in the following Fig. 4, and the algorithm of restarting PINN-Adam, or r-PINN-Adam, is presented in Algorithm 1.

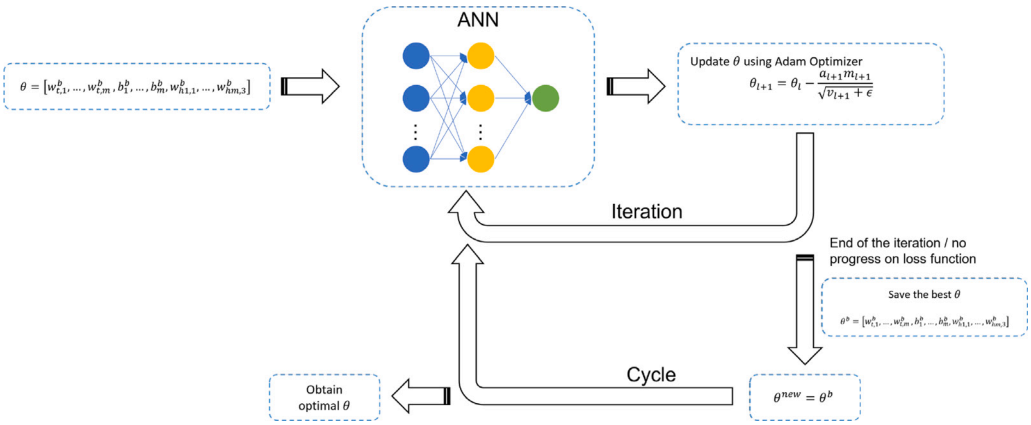



**Fig. 4.** Restarting process of PINN-Adam.

---

**Algorithm 1** r-PINN with Adam optimizer

1: Define a mathematical model system of ODE population dynamic;
2: Define the ODE points ($t_i$), and initial points ($t_0$);
3: Determine the number of training points from $t_i$;
4: Designing ANN architecture by specifying the number of layers and neurons;
5: **while** $c = 0 \leq$ max cycle **do**
6:     **while** $l = 0 \leq$ max iteration **do**
7:         Set $\theta$ with normal distribution in input and hidden layer, and random uniform in output layer.;
8:         Obtain output $\hat{x}(t_i;\theta)$, $\hat{y}(t_i;\theta)$ and $\hat{z}(t_i;\theta)$ using Equation (2)–(4);
9:         Compute the derivative of $\hat{x}(t_i;\theta)$, $\hat{y}(t_i;\theta)$ and $\hat{z}(t_i;\theta)$ using AD as in Equation (5);
10:         Compute the loss function by using current $\theta^l$

$$\mathcal{L}(\theta^l) = MSE_{x0} + MSE_{y0} + MSE_{z0} + MSE_x + MSE_y + MSE_z;$$

11:         Update $\theta^l$ using Adam optimizer as in Equation (8)–(10);
12:         Set $t$ number of subsequent iteration to monitor;
13:         Monitor last $t$ iteration;
14:         **if** $\min[\mathcal{L}(\theta_l), \mathcal{L}(\theta_{l-1}, \dots, \mathcal{L}(\theta_{l-t})) = \mathcal{L}(\theta_{l-t})]$ **then**
15:             Save $\theta_{l-t}$ as $\theta^b$
16:             **break**;
17:         **else**
18:             Save $\theta_l$ as $\theta^b$;
19:         **end if**
20:         $l = l + 1$;
21:     **end while**
22:     Load $\theta^b$ and set as initiation of ANN trainable parameter;

$$\theta^{new} = \theta^b = [w^b_{t,1}, \dots, w^b_{t,m}, b^b_1, \dots, b^b_m, w^b_{h1,1}, \dots, w^b_{hm,3}];$$

23:     Restart from step 8 where $\hat{x}(t_i;\theta^{new})$, $\hat{y}(t_i;\theta^{new})$, and $\hat{z}(t_i;\theta^{new})$;
24:     $c = c + 1$;
25: **end while**
26: Obtained $\theta^b$ as the best $\theta$ based on the minimum loss function value;
27: Load $\theta^b$ to applied in the testing points;
28: Obtained approximate solution for the entire domain;

---

## 4. Results and discussion

The r-PINN algorithm with the Adam optimizer is applied to predict the population dynamics system of sea turtles, as described in Eq. (1). We validate our method using the solution obtained by RK4 method for these nonlinear models. The code for this method was written in Python and executed on a Windows 11 laptop equipped with 8 GB RAM, an Intel Core i7-8565U processor, and an Intel UHD 620 graphics card. The total number of ODE points is set to 2000, with 80% of these points designated as training points.

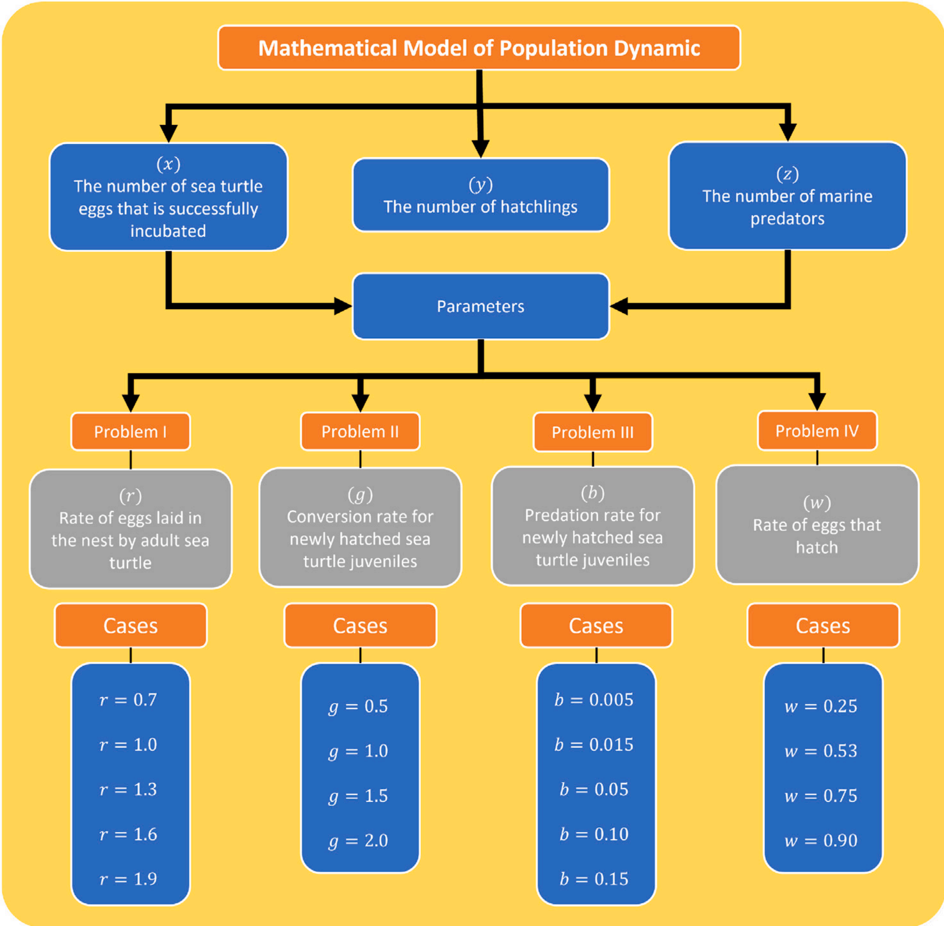



**Fig. 5.** Graphical illustration of this study.

**Table 2**
Population dynamic of sea turtles parameters values for all problems.

| Parameters | Problem I ($r$) | Problem II ($g$) | Problem III ($b$) | Problem IV ($w$) |
|---|---|---|---|---|
| $r$ | Simulated | 1.3 | 1.3 | 1.3 |
| $K$ | 700 | 700 | 700 | 700 |
| $a$ | 0.2 | 0.2 | 0.2 | 0.2 |
| $w$ | 0.53 | 0.53 | 0.53 | Simulated |
| $c$ | 0.001 | 0.001 | 0.001 | 0.001 |
| $b$ | 0.015 | 0.015 | Simulated | 0.015 |
| $g$ | 1 | Simulated | 1 | 1 |
| $s$ | 0.7 | 0.7 | 0.7 | 0.7 |
| $f$ | 1 | 1 | 1 | 1 |

Furthermore, these training points are further divided, with 15% set aside for validation. The ANN architecture is constructed with 5 hidden layers, each containing 100 neurons. The maximum number of iterations is set to 100, and this process is repeated within 30 cycles. The initial values applied for $x(0)$, $y(0)$, and $z(0)$ are 1000, 100, and 10, respectively. We consider four problems for different cases, depending on the values of the parameters, i.e., $r$, $g$, $b$, and $w$. All of the problems, as well as their cases, are illustrated in Fig. 5, while the values of all parameters are presented in Table 2.

In addition to MSE, the performance metrics for the proposed method are also evaluated in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The mathematical formulation for the approximation model of the population dynamics, in the case of RMSE and MAE, is presented as follows:

$$\begin{cases} RMSE_{x0} = \sqrt{\left[\hat{x}(t_0;\theta) - x(t_0)\right]^2}, \\ RMSE_{y0} = \sqrt{\left[\hat{y}(t_0;\theta) - y(t_0)\right]^2}, \\ RMSE_{z0} = \sqrt{\left[\hat{z}(t_0;\theta) - z(t_0)\right]^2}, \\ RMSE_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t_i}\hat{x}(t_i;\theta) - r\hat{x}(t_i;\theta)\left(1 - \frac{\hat{x}(t_i;\theta)}{K}\right) + a\hat{x}(t_i;\theta) + w\hat{x}(t_i;\theta)\right]^2}, \\ RMSE_y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t_i}\hat{y}(t_i;\theta) - gw\hat{x}(t_i;\theta) + c\hat{y}(t_i;\theta) + b\hat{y}(t_i;\theta)\hat{z}(t_i;\theta)\right]^2}, \\ RMSE_z = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t_i}\hat{z}(t_i;\theta) - fb\hat{y}(t_i;\theta)\hat{z}(t_i;\theta) + s\hat{z}(t_i;\theta)\right]^2}. \end{cases} \tag{11}$$

$$\begin{cases} MAE_{x0} = \left|\hat{x}(t_0;\theta) - x(t_0)\right|, \\ MAE_{y0} = \left|\hat{y}(t_0;\theta) - y(t_0)\right|, \\ MAE_{z0} = \left|\hat{z}(t_0;\theta) - z(t_0)\right|, \\ MAE_x = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\partial}{\partial t_i}\hat{x}(t_i;\theta) - r\hat{x}(t_i;\theta)\left(1 - \frac{\hat{x}(t_i;\theta)}{K}\right) + a\hat{x}(t_i;\theta) + w\hat{x}(t_i;\theta)\right|, \\ MAE_y = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\partial}{\partial t_i}\hat{y}(t_i;\theta) - gw\hat{x}(t_i;\theta) + c\hat{y}(t_i;\theta) + b\hat{y}(t_i;\theta)\hat{z}(t_i;\theta)\right|, \\ MAE_z = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\partial}{\partial t_i}\hat{z}(t_i;\theta) - fb\hat{y}(t_i;\theta)\hat{z}(t_i;\theta) + s\hat{z}(t_i;\theta)\right|. \end{cases} \tag{12}$$

### 4.1. Problem I: Analyzing the impact of variations in $r$ on the population dynamic model

In this study, we discuss the impact of variations in the rate of eggs laid in the nest by adult sea turtle ($r$). As depicted in Fig. 5, we consider five different cases for the parameter $r$. A comparison between the approximate solutions derived from the r-PINN-Adam algorithm and the RK4 method is illustrated in Fig. 6.

In Fig. 7, the range of values for weights and biases from the input layer to hidden layer 4 is set between −4 and 4. For hidden layer 5, however, we set the range of values for weights and biases between 0 and 50 for cases 1, 2, and 3 respectively, from 0 to 75 for case 4, and from 0 to 100 for case 5. We treat the hidden layers differently due to the requirements of the output. Consequently, after the training process, we obtain the optimal weights and biases within these ranges.

On the left side of Fig. 8 shows the loss function value history which includes MSE, RMSE, and MAE of training and validation points. Meanwhile, the right side shows the smallest loss function value obtained in each cycle for all cases. It should be noted that r-PINN only monitors the validation MSE, if the validation MSE does not improve in the next few iterations, the process will be stopped and restart to the next cycle.

We also simulated the value of each initial condition shown in Fig. 9. We used values ranged from 1000 to 5000 for $x(0)$, from 100 to 1000 for $y(0)$, and from 10 to 100 for $z(0)$.

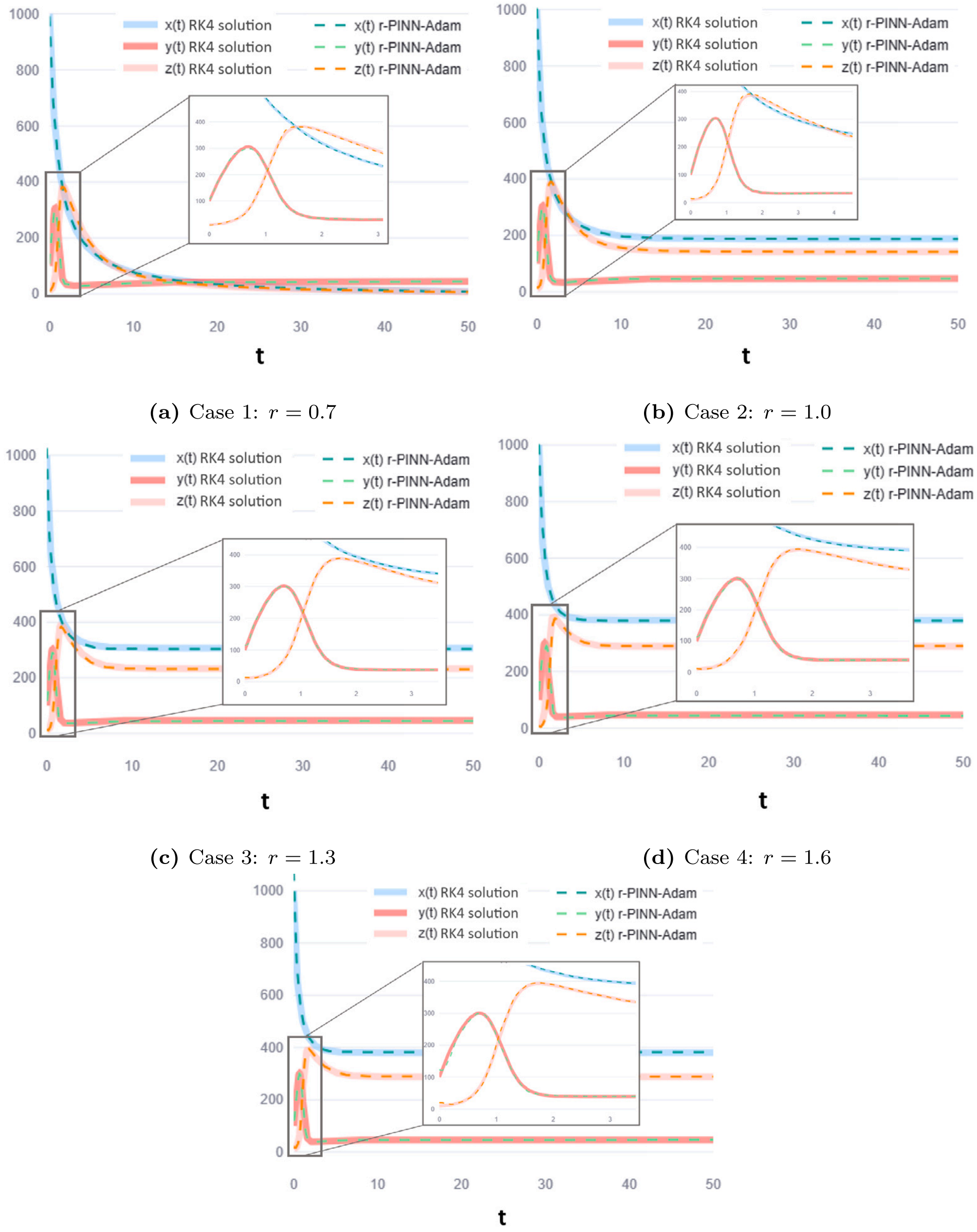
Fig. 10 presents a comparison of ($r$), which indicates the rate of eggs laid in the nest. We used the initial conditions 1000, 100, and 10 for $x(0)$, $y(0)$, and $z(0)$, respectively. As can be observed from Fig. 10(a), a larger value of $r$ corresponds to a greater number of eggs that can survive predation ($x(t)$). Similarly, as shown in Fig. 10(b), an increase in $r$ also increases the number of hatchling eggs ($y(t)$), although the improvement is not substantial. Lastly, a larger rate of $r$ also leads to an increase in the number of predators ($z(t)$), as seen in Fig. 10(c).

Based on the results obtained from the various values of $r$ in Problem I, it can be deduced that a positive correlation exists between $x(t)$ and $z(t)$ with respect to $r$. Moreover, while an increase in the number of $y(t)$ is observed, its direct relationship with $r$ is not significantly evident. Consequently, the conservation of the sea turtle population depends on the increase of $r$. This objective can be achieved through the implementation of strategies such as protecting nesting sites from predators, regulating human activities near sea turtle nesting areas, and closely monitoring the predator population density. Furthermore, efforts can be made to raise public awareness about the importance of sea turtle conservation and promote sustainable tourism practices that minimize negative impacts on sea turtle habitats.

Additionally, Table 3 offers a comprehensive overview of the results illustrated in Figs. 7 through 10. These overview include key metrics such as the total number of iterations, cycles, time, as well as specific values of the loss functions MSE, RMSE, and MAE. It is noteworthy that the cumulative count of training iterations for 30 cycles varies due to the fluctuating number of iterations within each case across cycles.

### 4.2. Problem II: Analyzing the impact of variations in $g$ on the population dynamic model

In this problem, we discuss the impact of variations in the conversion rate from eggs to hatchlings ($g$). Four distinct cases are examined, as depicted in Fig. 5. The results of the approximate solutions across all cases are presented in Fig. 11. Additional findings




**(a)** Case 1: $r = 0.7$

**(b)** Case 2: $r = 1.0$

**(c)** Case 3: $r = 1.3$

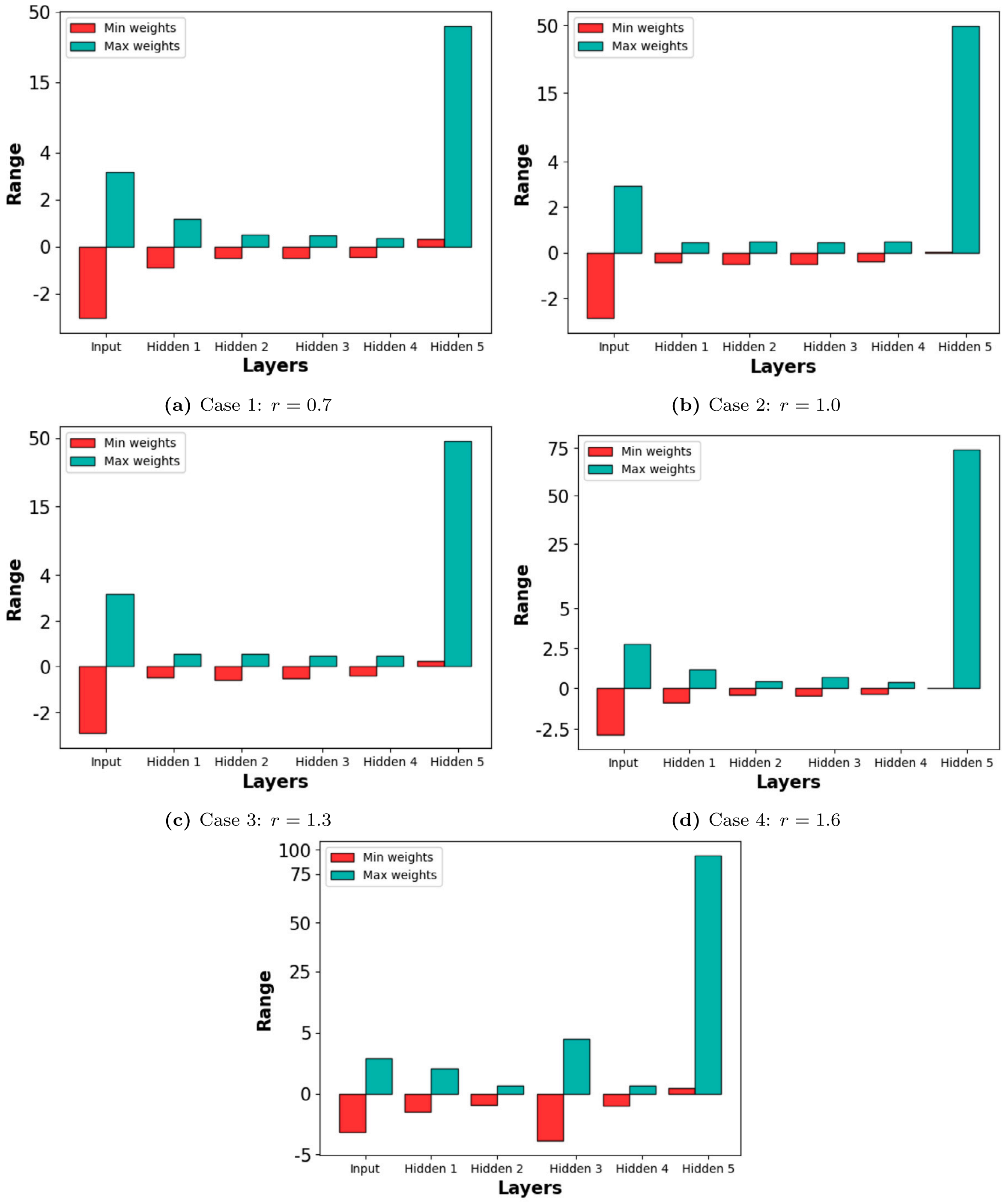
**(d)** Case 4: $r = 1.6$

**(e)** Case 5: $r = 1.9$

**Fig. 6.** Comparison of RK4 solutions and approximate solutions based on r-PINN-Adam for all cases of $r$.

are illustrated in Figs. 12 and 13, which respectively portray the distribution of trainable parameters, consisting of weights and biases, and a history of the loss function value and minimum loss values in each cycle of r-PINN-Adam for all $g$ variants. Further insights are derived from trajectory simulations for initial conditions: $x(0)$ ranging from 1000 to 5000, $y(0)$ ranging from 100 to 1000, and $z(0)$ ranging from 10 to 100, as displayed in Fig. 14. Moreover, Fig. 15 exhibits the behavioral patterns of $x(t)$, $y(t)$, and $z(t)$ within the population dynamics, influenced by all $g$ cases.

Fig. 15 presents the findings of Problem II on the population dynamics system of sea turtles. Initially, the variations in the conversion rate from eggs to hatchlings ($g$) have no impact on the population density of sea turtle eggs ($x(t)$). This can be observed in Fig. 15(a), which shows no change in $x(t)$ for any value of $g$. Additionally, Fig. 15(b) demonstrates a strong positive relationship between $g$ and the population density of hatchling eggs ($y(t)$), as the number of hatchlings increases significantly. However, due to




(a) Case 1: $r = 0.7$

(b) Case 2: $r = 1.0$

(c) Case 3: $r = 1.3$

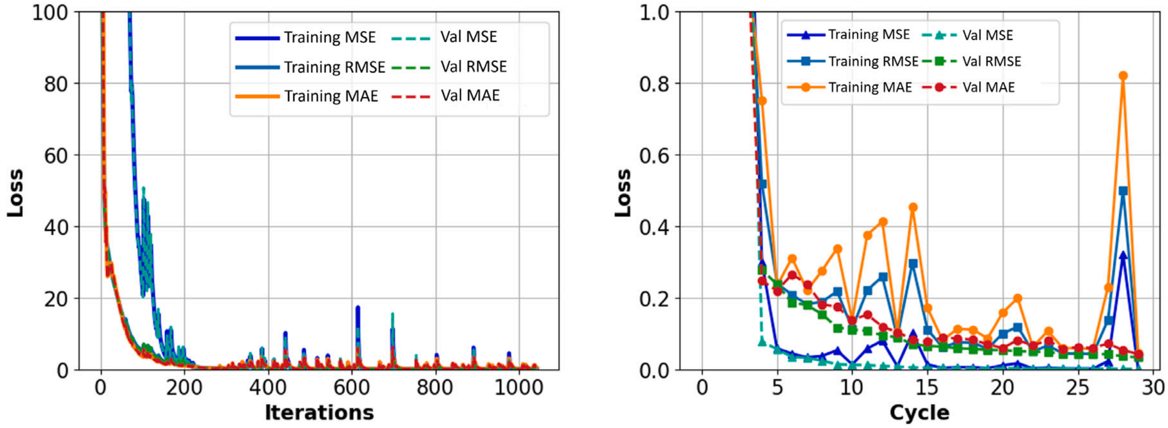
(d) Case 4: $r = 1.6$

(e) Case 5: $r = 1.9$

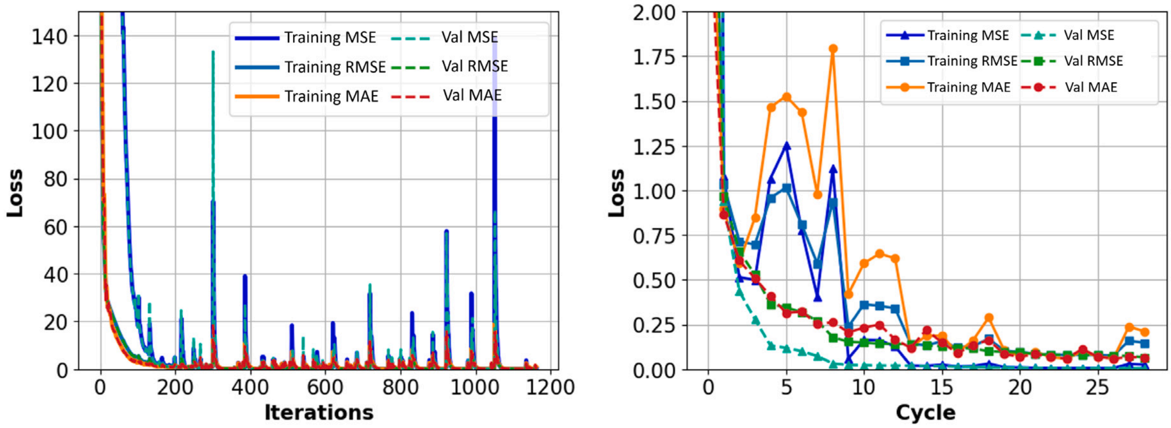
**Fig. 7.** The best weights and biases range distribution in each layer for all case variations of $r$.

the rapid increase in the number of predators ($z(t)$) depicted in Fig. 15(c), the number of $y(t)$ poses a threat to the sustainability of the population.

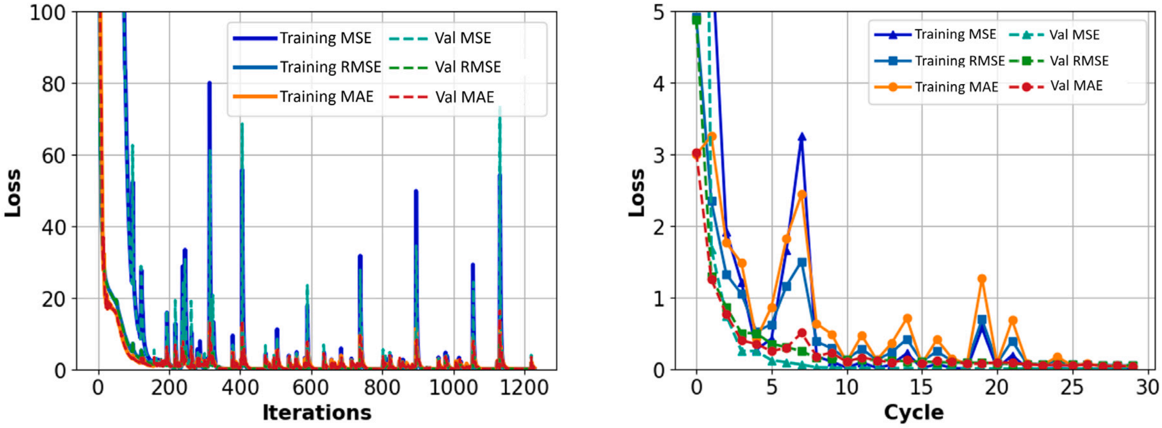
In conclusion, to maintain the population of sea turtle hatchlings, efforts should be directed towards reducing the number of predators. This can be accomplished by implementing measures such as increasing the number of nest protectors, regulating fishing




**(a)** Case 1: $r = 0.7$




**(b)** Case 2: $r = 1.0$



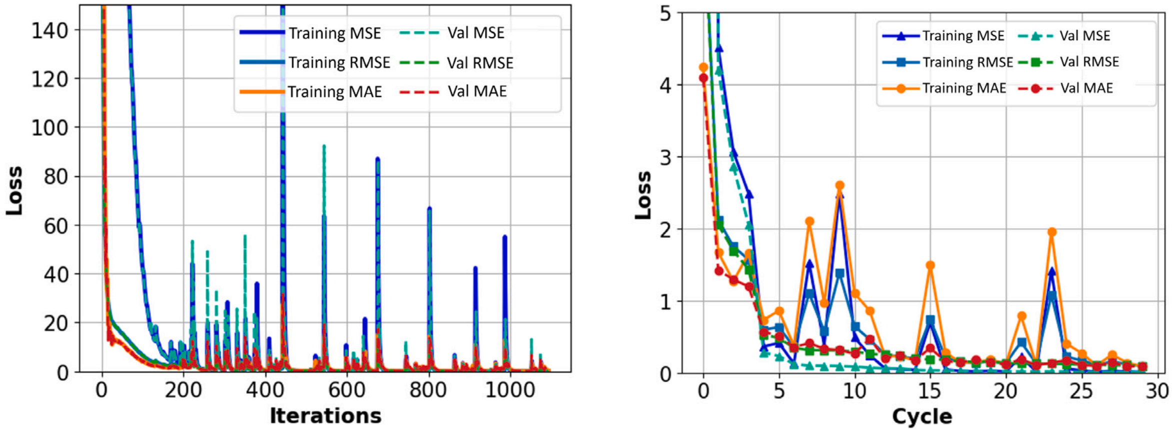

**(c)** Case 3: $r = 1.3$

**Fig. 8.** Performance of r-PINN-Adam: loss function (left) and the minimum loss function values for each cycle (right) for all case variations of $r$.

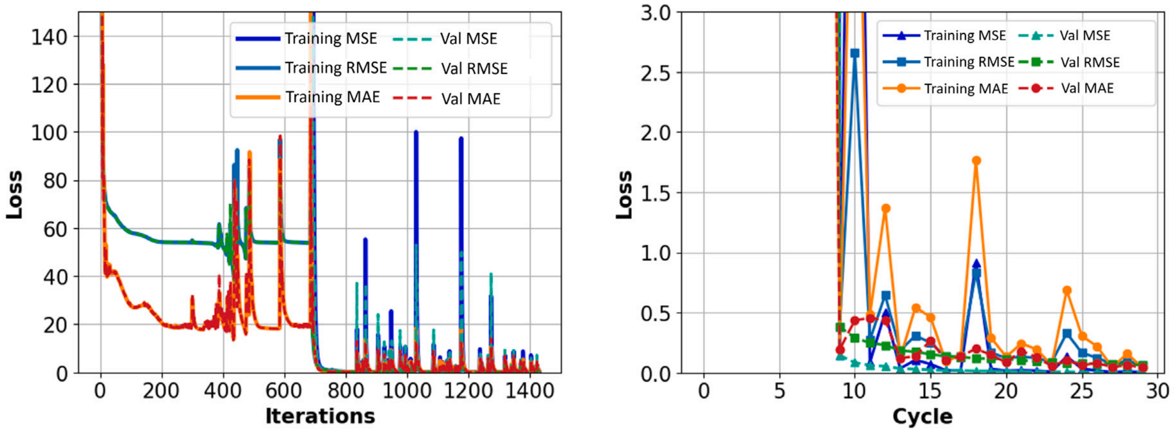
activities in proximity to sea turtle nesting areas, and mitigating light pollution near beaches. Lastly, Table 4 provides a summary of the results, displaying the total iterations, cycles, time, and specific values of the loss functions MSE, RMSE, and MAE.

### *4.3. Problem III: Analyzing the impact of variations of $b$ on the population dynamic model*

Five cases are considered to investigate the effect of variations in the predation rate for hatchlings by marine predators ($b$). Fig. 16 illustrates the comparison between the RK4 solutions and the approximate solutions obtained by the r-PINN-Adam algorithm for




**(d)** Case 4: $r = 1.6$



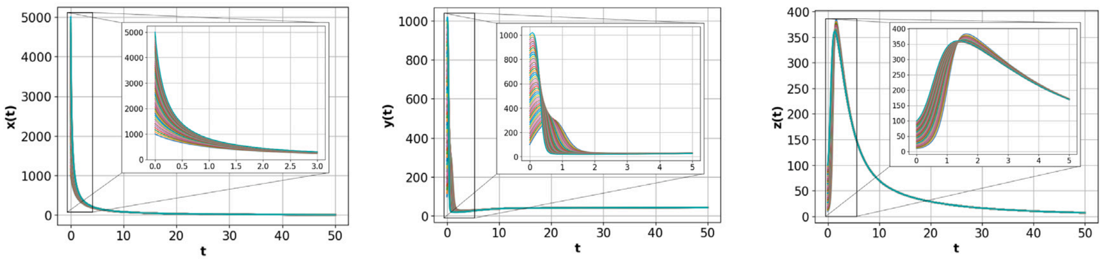

**(e)** Case 5: $r = 1.9$

**Fig. 8.** (*continued*).

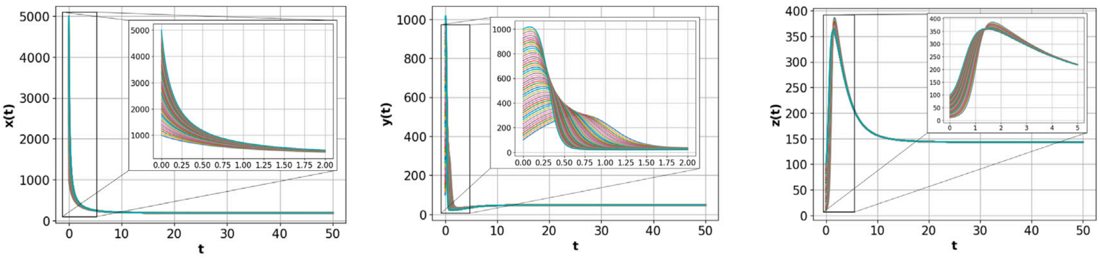
**Table 3**
Results summary of r-PINN-Adam performance in the variations of $r$.

| Case | Iteration | Cycle | Time/iter | Total time | Data type | MSE | RMSE | MAE |
|---|---|---|---|---|---|---|---|---|
| I | 1043 | 30 | 2.72 s | 2839.223 s | Training<br>Validation<br>Testing | 1.137e−3<br>1.101e−3<br>1.101e−3 | 3.369e−2<br>3.318e−2<br>3.317e−2 | 4.633e−2<br>4.314e−2<br>4.313e−2 |
| II | 1163 | 30 | 2.64 s | 3075.047 s | Training<br>Validation<br>Testing | 2.489e−2<br>4.355e−3<br>4.354e−3 | 1.472e−1<br>6.599e−2<br>6.601e−2 | 2.1073-1<br>6.511e−1<br>6.515e−1 |
| III | 1129 | 30 | 2.71 s | 3335.617 s | Training<br>Validation<br>Testing | 2.349e−3<br>2.370e−3<br>2.372e−3 | 4.847e−2<br>4.868e−2<br>4.868e−2 | 3.462e−2<br>3.764e−2<br>3.763e−2 |
| IV | 1093 | 30 | 2.62 s | 2858.568 s | Training<br>Validation<br>Testing | 8.920e−3<br>8.763e−3<br>8.763e−3 | 9.441e−2<br>9.362e−2<br>9.361e−2 | 8.707e−2<br>8.622e−2<br>8.623e−2 |
| V | 1430 | 30 | 2.63 s | 3764.288 s | Training<br>Validation<br>Testing | 4.074e−3<br>4.071e−3<br>4.081e−3 | 6.382e−2<br>6.380e−2<br>6.388e−2 | 4.075e−2<br>4.249e−2<br>4.248e−2 |

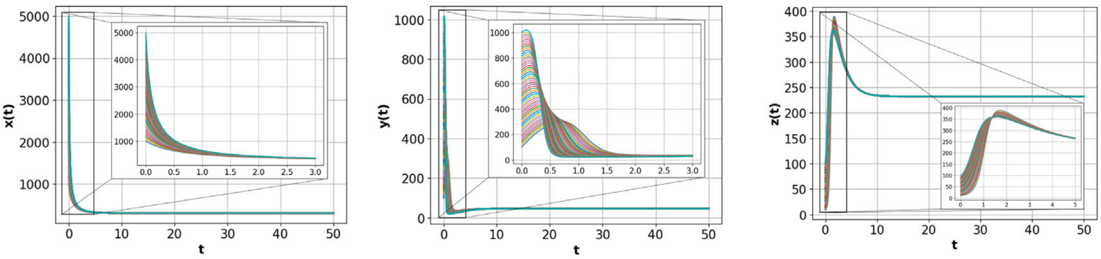
all five cases. The minimum and maximum values of $\theta$ that yield the best solution for each layer across all cases are depicted in Fig. 17. The history of the loss function value and the smallest loss function value obtained in each cycle for all cases are visualized in Fig. 18. The initial condition simulation values range from 1000 to 5000 for $x(0)$, from 100 to 1000 for $y(0)$, and from 10 to 100



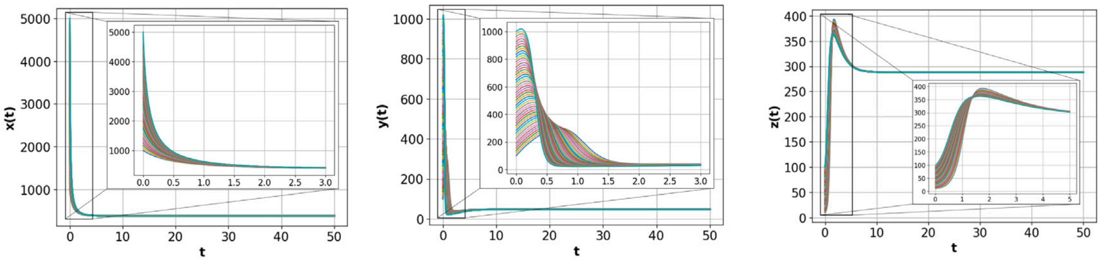

**(a)** Case 1: $r = 0.7$



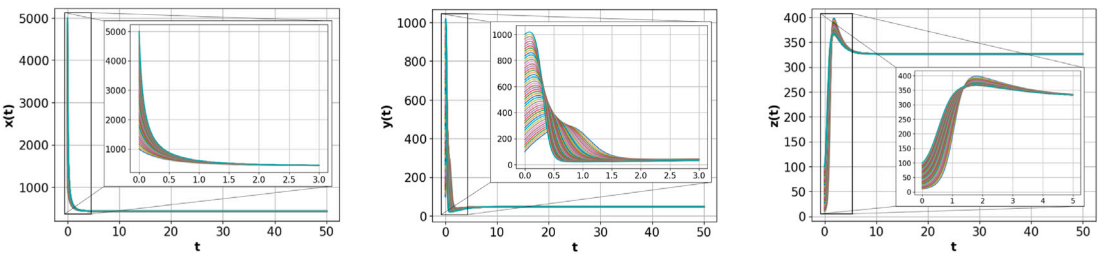

**(b)** Case 2: $r = 1.0$




**(c)** Case 3: $r = 1.3$

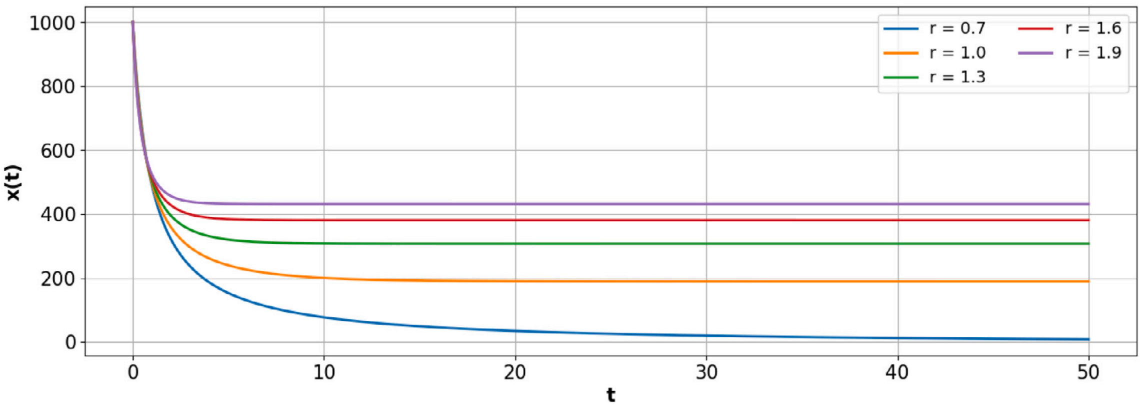



**(d)** Case 4: $r = 1.6$

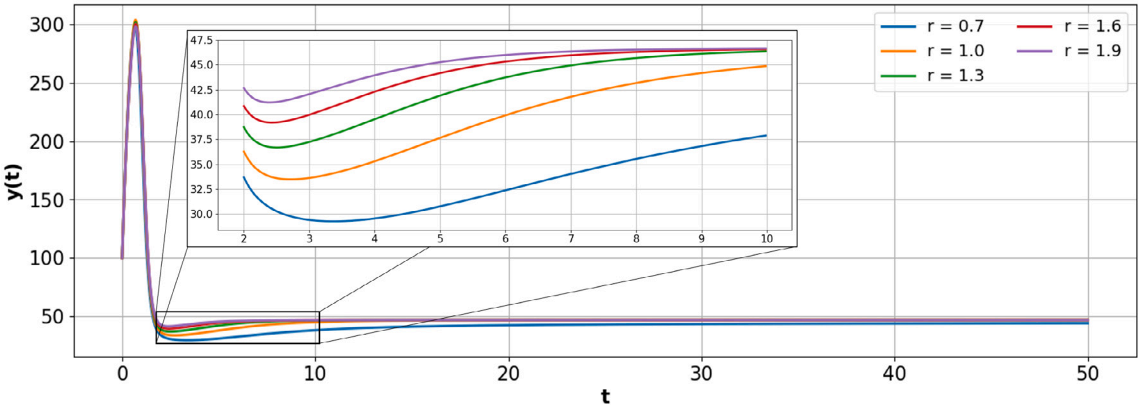



**(e)** Case 5: $r = 1.9$

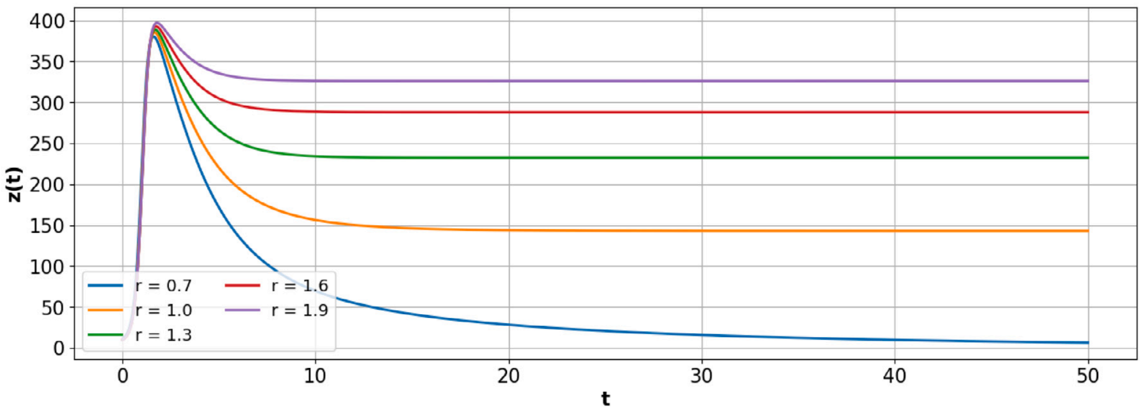
**Fig. 9.** The trajectory of $x(0)$ (left), $y(0)$ (mid), and $z(0)$ (right) for all variations of $r$.




**(a)** $x(t)$




**(b)** $y(t)$

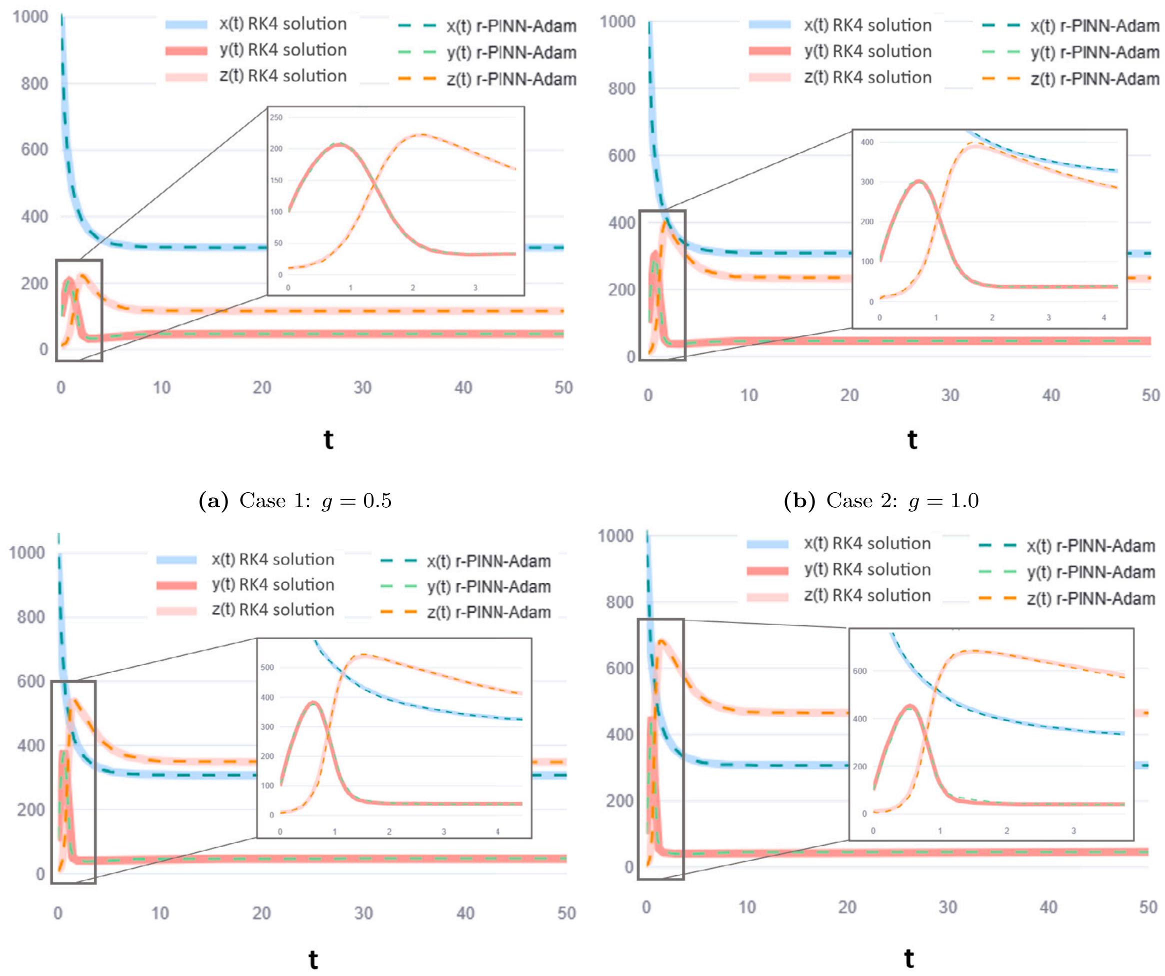



**(c)** $z(t)$

**Fig. 10.** Approximate solutions of $x(t), y(t)$ and $z(t)$ using r-PINN-Adam algorithm with the variation values of $r$ on the population dynamic model of sea turtles.

for $z(0)$, as shown in Fig. 19. Fig. 20 displays the behavior of each $x(t)$, $y(t)$, and $z(t)$ for the population dynamic model under the influence of all $b$ cases.

Fig. 20 illustrates the relationship between the variation in the predation rate for hatchlings by marine predators ($b$) and the three variables: the population density of sea turtle eggs ($x(t)$), the population density of hatchling eggs ($y(t)$), and the population density of turtle predators ($z(t)$). In Fig. 20(a), it is observed that an increase in the value of $b$ has no effect on $x(t)$. However, the change in $b$ significantly affects $y(t)$. As depicted in Fig. 20(b), a substantial increase in $y(t)$ can be seen at the early stages of $t$, especially when $b = 0.005$. In other hand, this also causes an increase in $z(t)$, as seen in Fig. 20(c), leading to a significant decrease




(a) Case 1: $g = 0.5$

(b) Case 2: $g = 1.0$

(c) Case 3: $g = 1.5$

(d) Case 4: $g = 2.0$

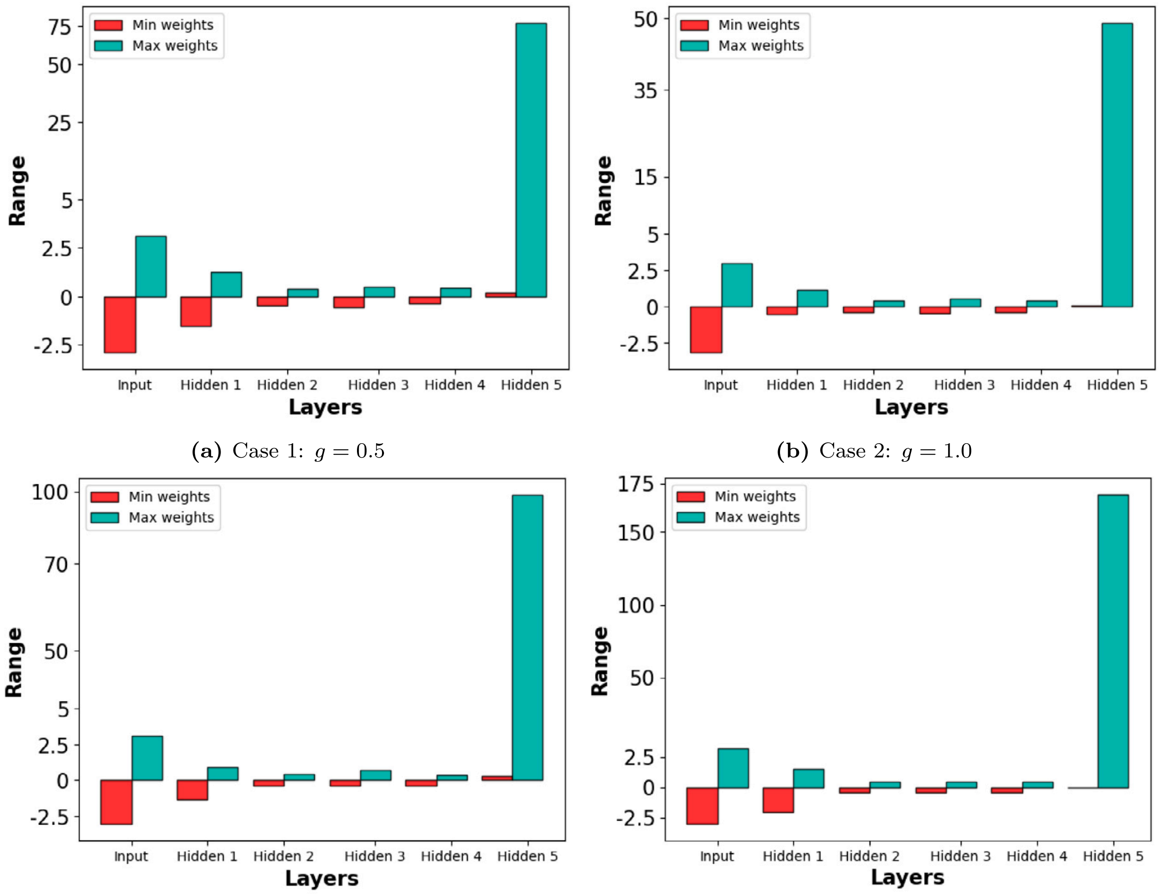
**Fig. 11.** Comparison of RK4 solutions and approximate solutions using r-PINN-Adam for all case variations of $g$.

**Table 4**
Results summary of r-PINN-Adam performance in the variations of $g$.

| Case | Iteration | Cycle | Time/iter | Total time | Data type | MSE | RMSE | MAE |
|---|---|---|---|---|---|---|---|---|
| I | 1025 | 30 | 2.55 s | 2622.844 s | Training | 6.371e−3 | 7.966e−2 | 9.579e−2 |
| | | | | | Validation | 5.955e−3 | 7.716e−2 | 7.635e−2 |
| | | | | | Testing | 5.955e−3 | 7.717e−2 | 7.638e−2 |
| II | 1104 | 30 | 2.51 s | 2771.132 s | Training | 2.247e−3 | 4.739e−2 | 4.730e−2 |
| | | | | | Validation | 2.223e−3 | 4.715e−2 | 4.571e−2 |
| | | | | | Testing | 2.225e−3 | 4.717e−2 | 4.572e−2 |
| III | 1323 | 30 | 2.62 s | 3475.254 s | Training | 2.579e−2 | 1.605e−1 | 1.269e−1 |
| | | | | | Validation | 2.570e−2 | 1.601e−1 | 1.253e−1 |
| | | | | | Testing | 2.569e−2 | 1.603e−1 | 1.254e−1 |
| IV | 1242 | 30 | 2.66 s | 3305.809 s | Training | 5.174e−2 | 2.227e−1 | 1.749e−1 |
| | | | | | Validation | 5.101e−2 | 2.260e−1 | 1.670e−1 |
| | | | | | Testing | 5.100e−2 | 2.258e−1 | 1.668e−1 |

in the number of $y(t)$ shortly thereafter. Furthermore, this relationship results in the population densities of $y(t)$ and $z(t)$ stabilizing at $t = 10$ and beyond.

In conclusion, it is crucial to maintain a low predation rate for hatchlings by marine predators to ensure the survival of sea turtles. Efforts can be made to enhance public awareness and education about the importance of sea turtle conservation and the




(a) Case 1: $g = 0.5$ (b) Case 2: $g = 1.0$ (c) Case 3: $g = 1.5$ (d) Case 4: $g = 2.0$

**Fig. 12.** The best weights and biases range distribution in each layer for all case variations of $g$.

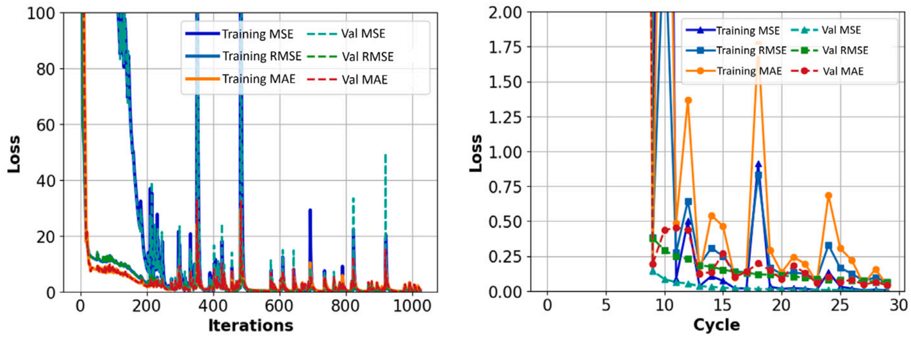
role that predators play in their survival. These measures can contribute to ensuring that future generations can enjoy the beauty and ecological significance of these remarkable creatures. Table 5 summarizes the results and presents the total iterations, cycles, time, and specific values of the loss functions MSE, RMSE, and MAE.

### 4.4. Problem IV: Analyzing the impact of variations in $w$ on the population dynamic model

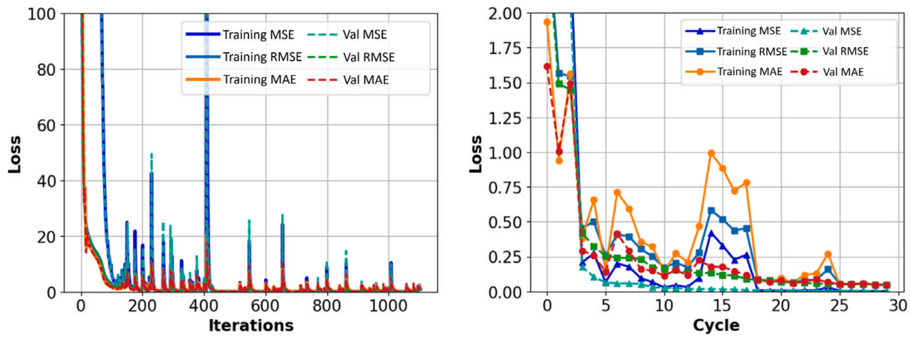
In this problem, we discuss four cases that depend on the variations in the successful rate of eggs to hatch ($w$). The proposed method is applied to the system of sea turtles, and the results are compared with the RK4 method as illustrated in Fig. 21. The trainable parameters obtained for all variants of $w$ are presented in Fig. 22. The loss function value obtained for each iteration and in each cycle for all cases are shown in Fig. 23. The initial condition simulation values for $x(0)$, $y(0)$, and $z(0)$ are displayed in Fig. 24. Lastly, the behavior of each $x(t)$, $y(t)$, and $z(t)$ for all variants of $w$ is visualized in Fig. 25.

The results from varying successful rate of eggs hatching ($w$) in Problem IV are shown in Fig. 25. In Fig. 25(a), it is observed that an increase in the value of $w$ leads to a decrease in the population density of eggs laid in the nest ($x(t)$). This, in turn, causes an increase in the population density of hatchling eggs ($y(t)$) at early value of $t$, as shown in Fig. 25(b). However, this number decreases steeply shortly thereafter due to an increase in the number of predators ($z(t)$), as depicted in Fig. 25(c).

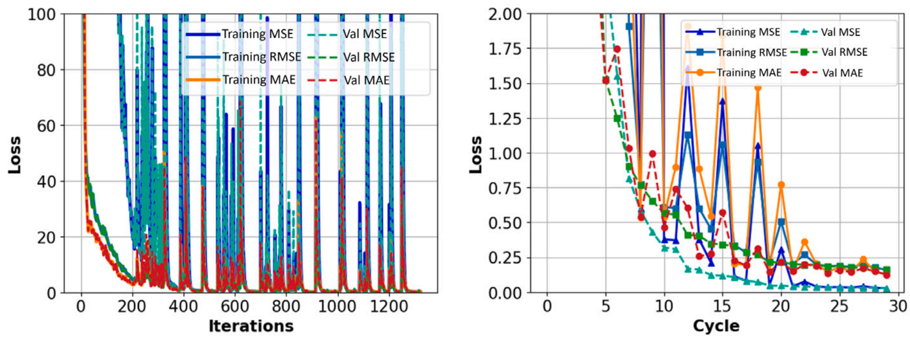
In conclusion, it is essential to protect nesting sites from predators and monitor predator population density. These efforts are crucial in ensuring that future generations can see the beauty and ecological significance of these remarkable creatures. Lastly, Table 6 provides a summary of all the results, including the total iterations, cycles, time, and the values of the loss functions MSE, RMSE, and MAE.



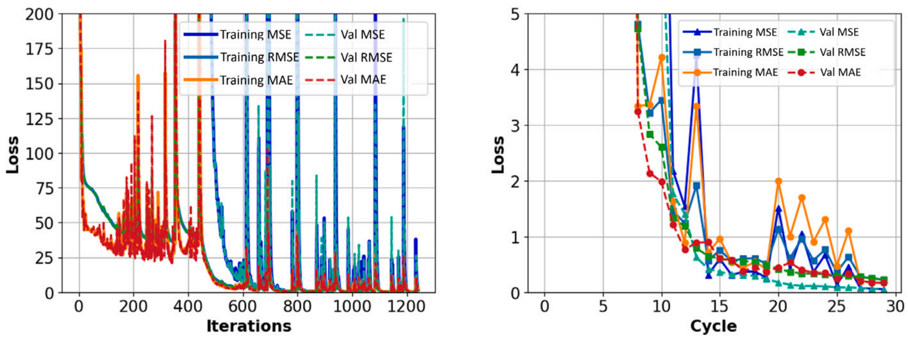

**(a)** Case 1: $g = 0.5$




**(b)** Case 2: $g = 1.0$

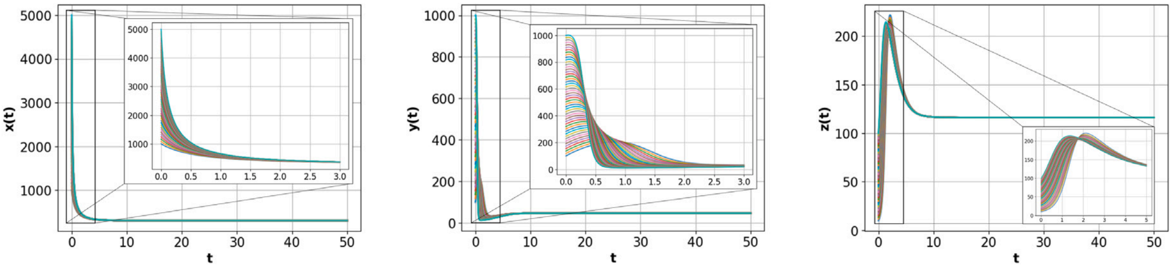



**(c)** Case 1: $g = 0.5$

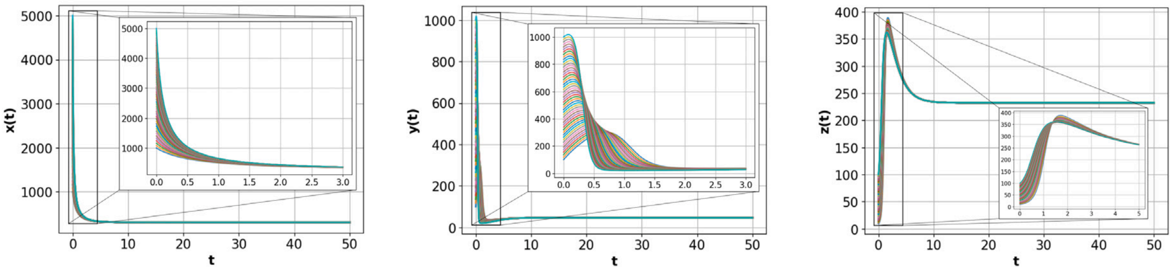



**(d)** Case 2: $g = 1.0$

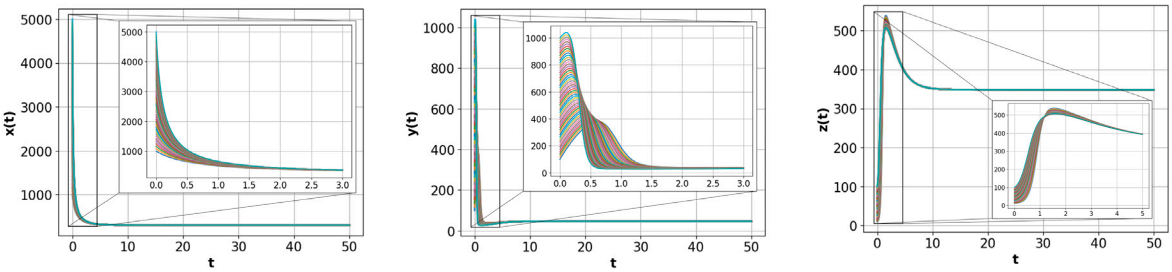
**Fig. 13.** Performance of r-PINN-Adam: loss function (left) and the minimum loss function values for each cycle (right) for all case variations of $g$.



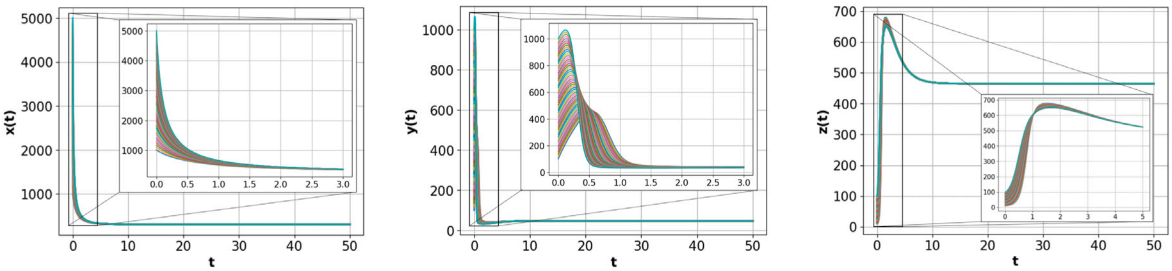

**(a)** Case 1: $g = 0.5$




**(b)** Case 2: $g = 1.0$




**(c)** Case 3: $g = 1.5$

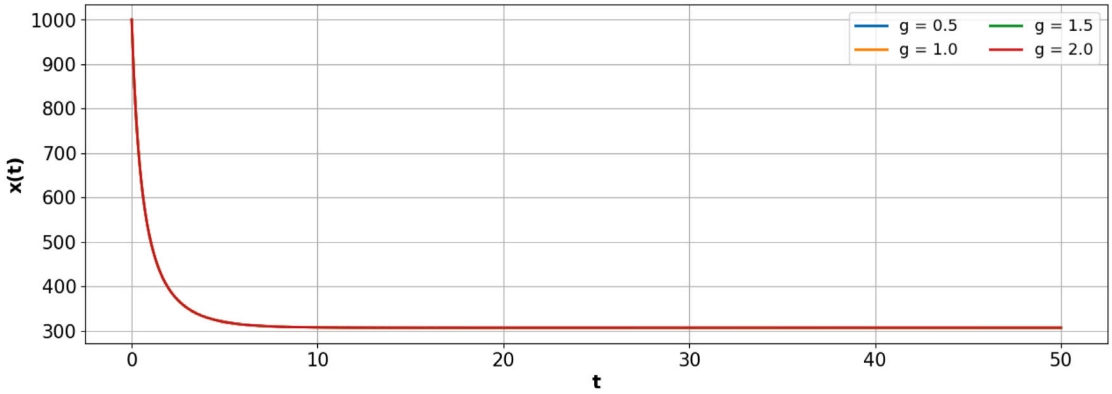



**(d)** Case 4: $g = 2.0$

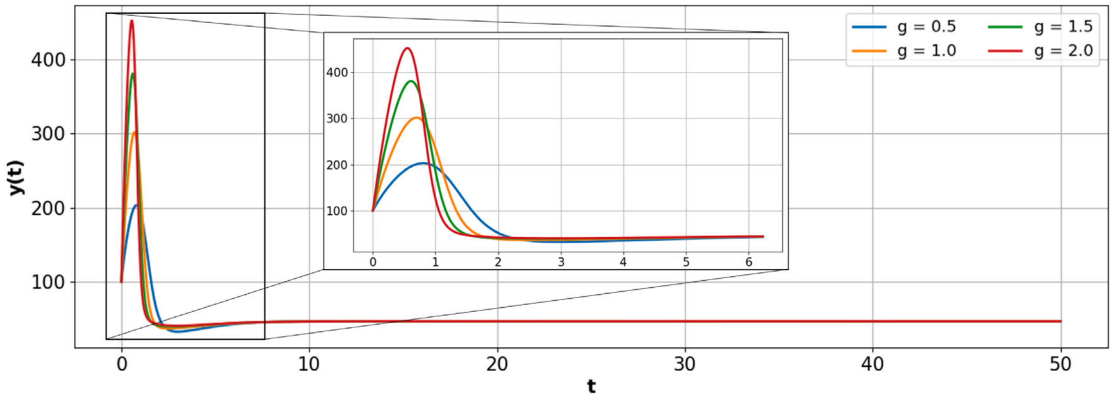
**Fig. 14.** The trajectory of $x(0)$ (left), $y(0)$ (mid), and $z(0)$ (right) for all case variations of $g$.

## 5. Conclusion

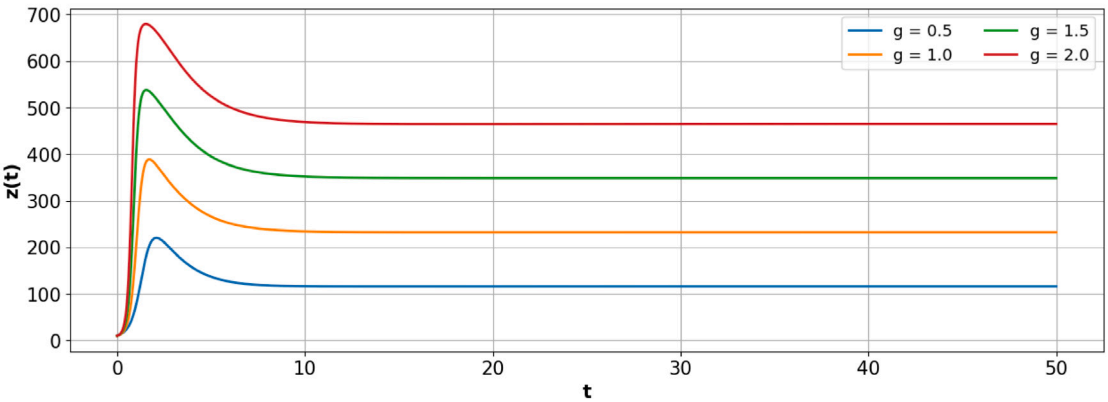
This study aims to investigated the Physic-Informed Neural Network (PINN) with restarting strategy combined with Adam optimizer (r-PINN-Adam) to solve the mathematical model of population dynamic system of sea turtles. The mathematical model is formulated with three compartments: the population density of eggs laid in the nest ($x$), the population density of hatchling eggs ($y$), and the population density of predators ($z$) as shown as in Eq. (1). Series of approximate solution is produced using the proposed r-PINN-Adam based on variations of parameters, including the eggs rate laid by a mother turtle ($r$), conversion rate from eggs to hatchlings ($g$), predation rate by a marine predator ($b$), and successful rate of eggs that hatched ($w$). Based on our analyze in the result, we summarize our findings as follows:




**(a)** $x(t)$



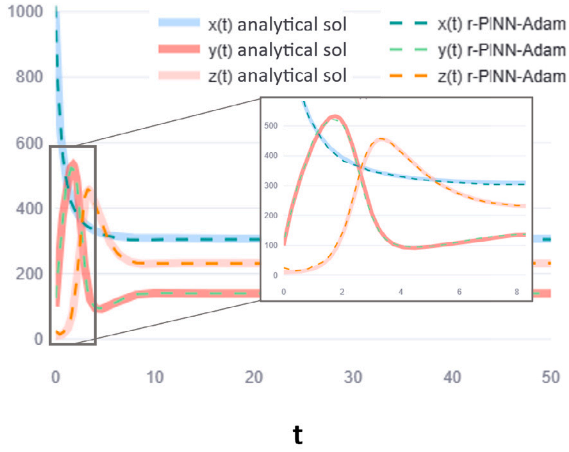

**(b)** $y(t)$



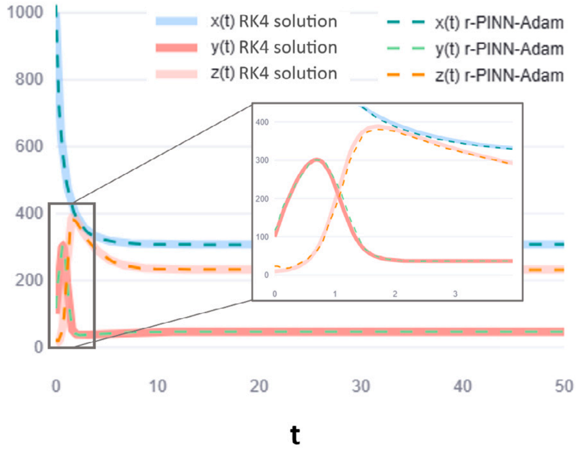

**(c)** $z(t)$

**Fig. 15.** Approximate solutions of $x(t)$, $y(t)$ and $z(t)$ using r-PINN-Adam algorithm with the variation values of $g$ on the population dynamic model of sea turtles.

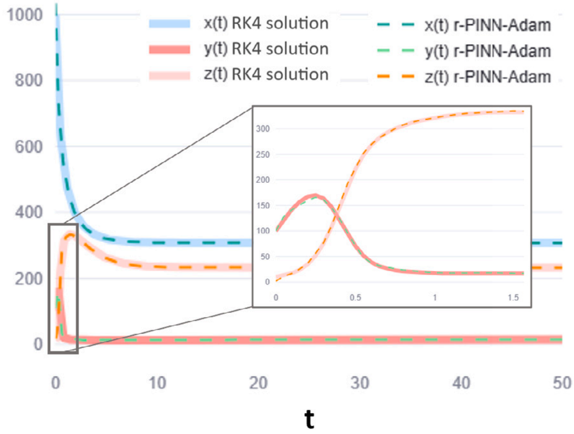
1. Variations in the parameters $r$, $g$, $b$, and $w$ were investigated for their impact on the output $x(t)$. The results showed a positive impact by the parameter $r$, while no significant impact was observed by $g$ and $b$, and a detrimental effect was observed by $w$.
2. The impact of variations in the parameters $r$, $g$, $b$, and $w$ on the outputs $y(t)$ and $z(t)$ was examined. Results indicated that variations in $r$, $g$, and $w$ had a positive impact on both outputs, whereas $b$ had an inverse impact on $y(t)$ and $z(t)$. These findings have implications for optimizing the system's performance.



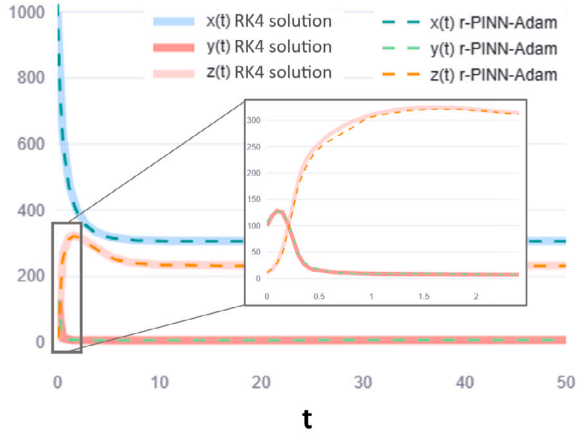

**(a)** Case 1: $b = 0.005$



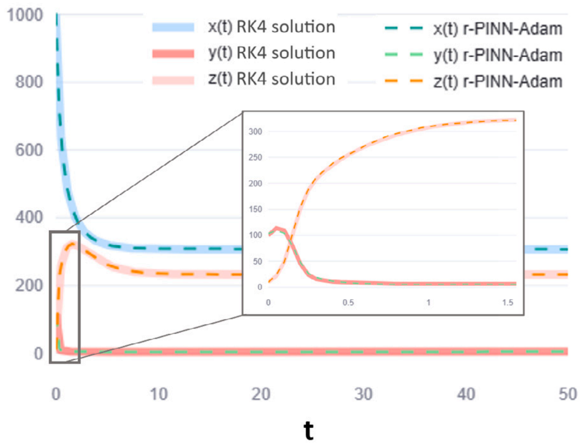

**(b)** Case 2: $b = 0.015$




**(c)** Case 3: $b = 0.05$

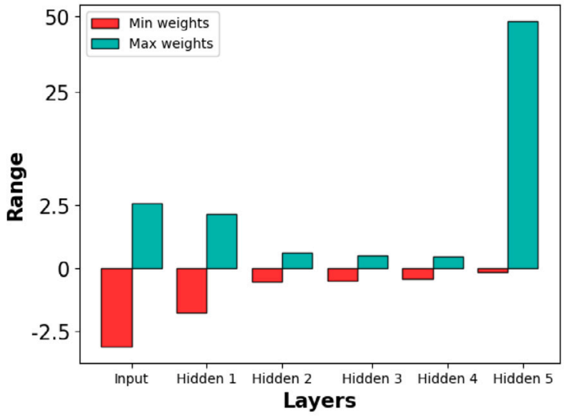



**(d)** Case 4: $b = 0.10$

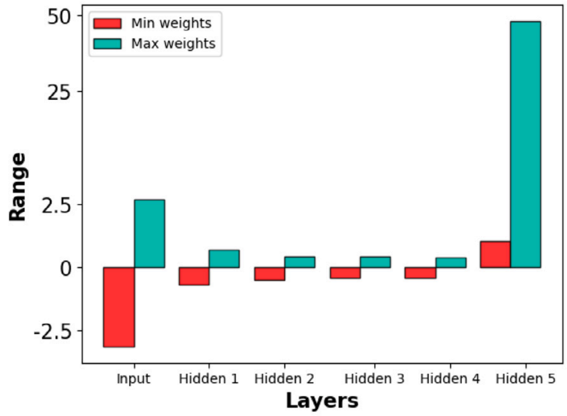



**(e)** Case 5: $b = 0.15$

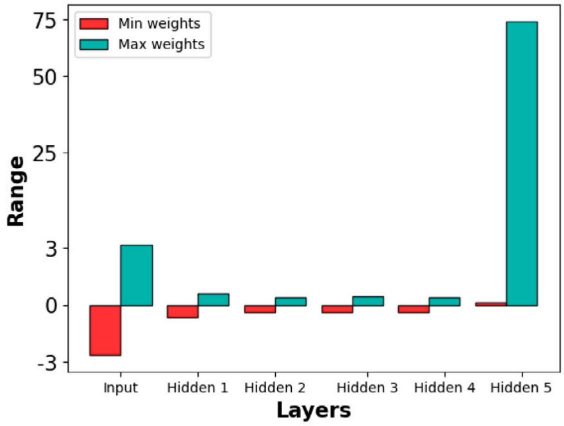
**Fig. 16.** Comparison of RK4 solutions and approximate solutions using r-PINN-Adam for all case variations of $b$.



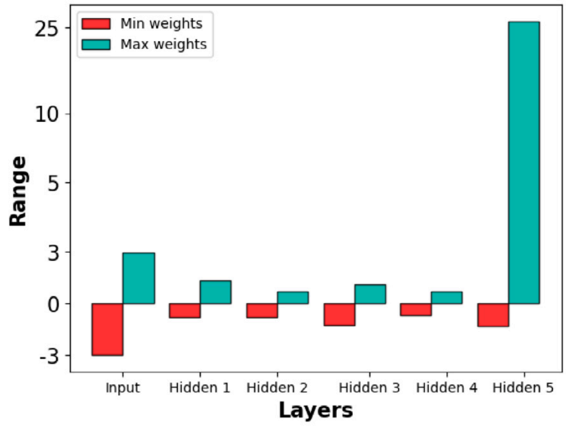

**(a)** Case 1: $b = 0.005$



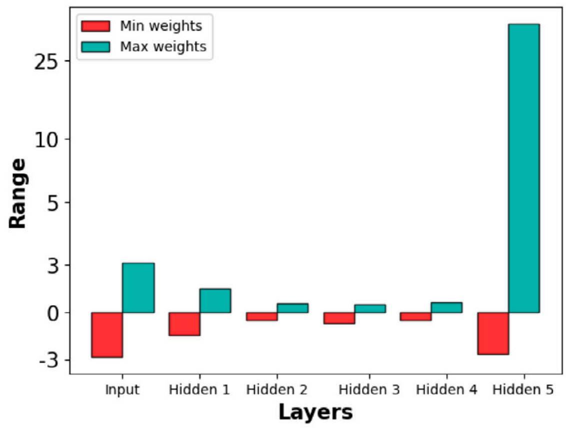

**(b)** Case 2: $b = 0.015$




**(c)** Case 3: $b = 0.05$



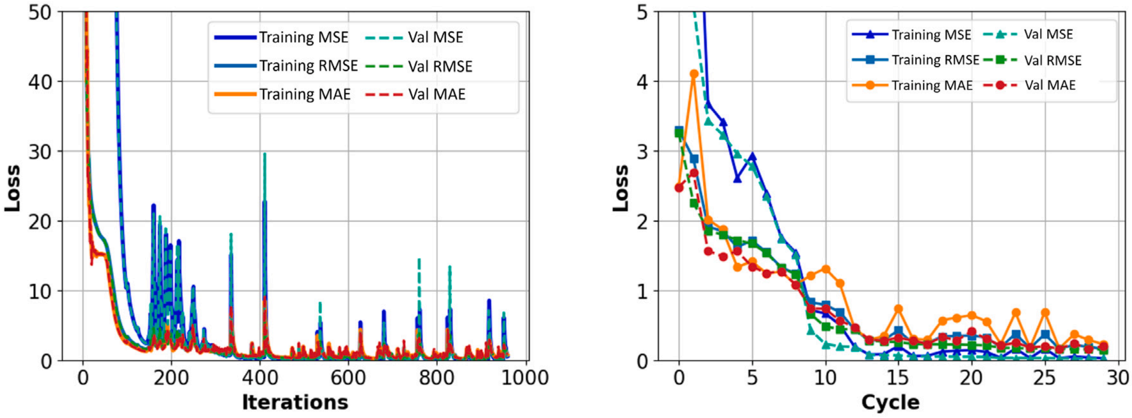

**(d)** Case 4: $b = 0.10$



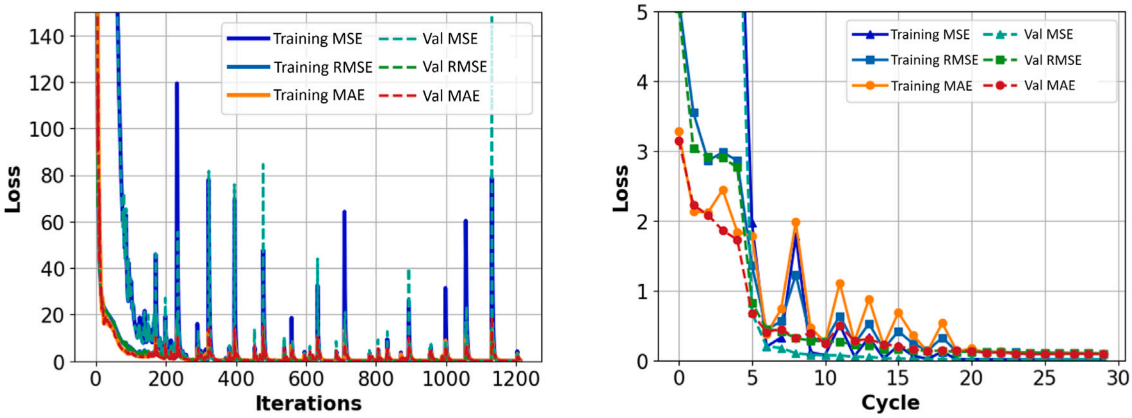

**(e)** Case 5: $b = 0.15$

**Fig. 17.** The best weights and biases range distribution in each layer for all case variations of $b$.

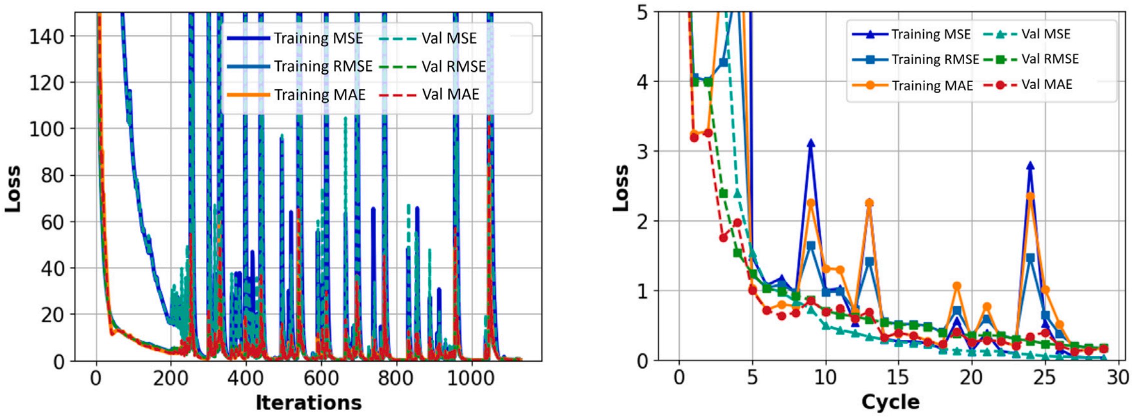
3. The loss function has been design as in Eq. (7) for MSE, and Eqs. (11), (12) for both RMSE and MAE, respectively. The results shows a good agreements for these loss function and can be seen in Figs. 8, 13, 18, and 23, and all summarize Tables 3, 4, 5, and 6.
4. The results obtained through r-PINN-Adam approximate solutions were found to be accurate compared to the solution obtained by RK4 method. The results can be seen in Figs. 6, 11, 16, and 21.




**(a)** Case 1: $b = 0.005$



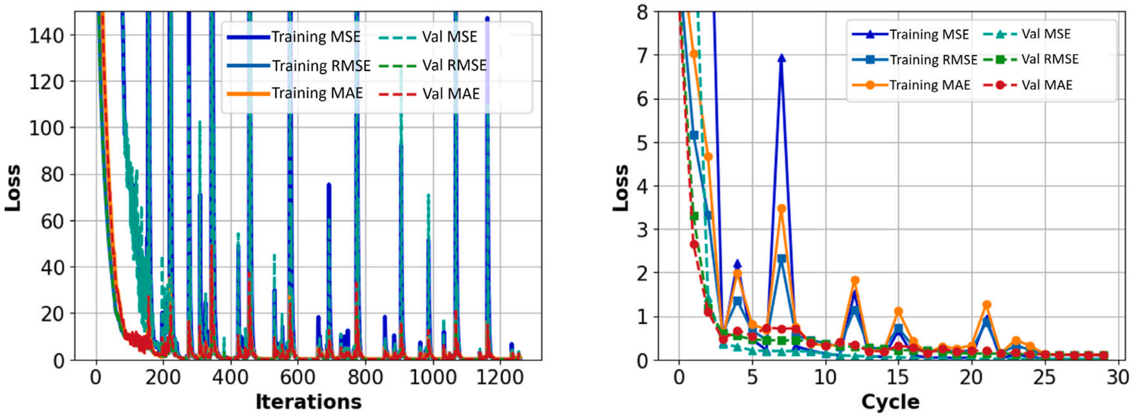

**(b)** Case 2: $b = 0.015$



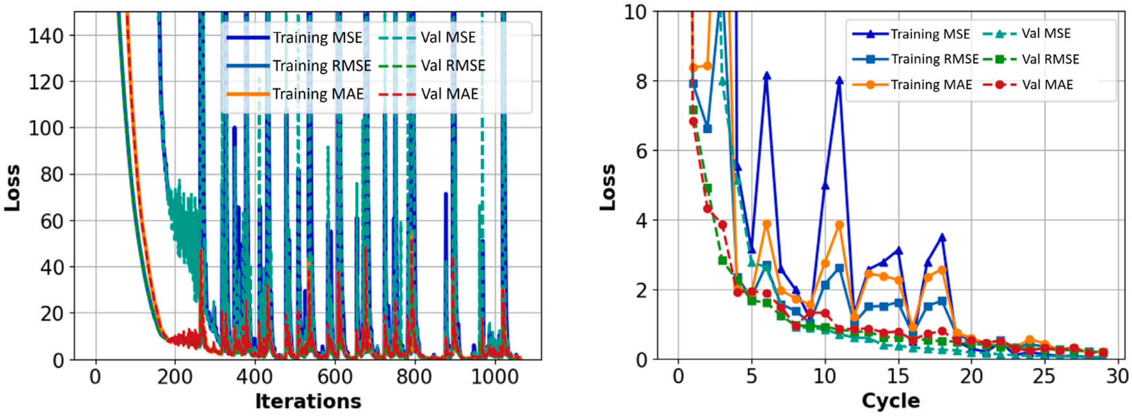

**(c)** Case 3: $r = b.0.05$

**Fig. 18.** Loss function value history (left) and loss function value minimum obtained in each cycle (right) for all case variations of $b$.

5. Based on our findings and the impact of variations in the $r$, $g$, $b$, and $w$ parameters, we suggest several actions that should be taken to preserve the sea turtle population. These actions include but are not limited to implementing measures to reduce pollution near beaches by regulating human activities near sea turtle nesting areas, protecting nesting and feeding habitats, monitoring the population density of predators, and raising public awareness about the importance of sea turtle conservation.
6. The proposed method also demonstrated excellent performance, as evidenced by its computation time and low values of performance metrics such as MSE, RMSE, and MAE, which are presented in summarize Tables 3, 4, 5, and 6. These findings




(d) Case 4: $b = 0.1$



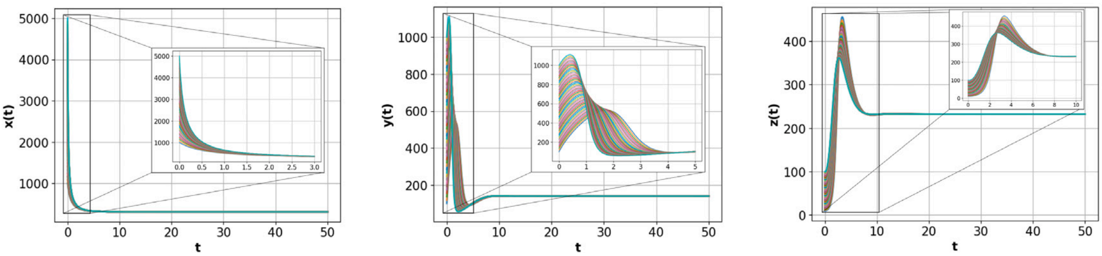

(e) Case 5: $b = 0.15$

**Fig. 18.** (*continued*).

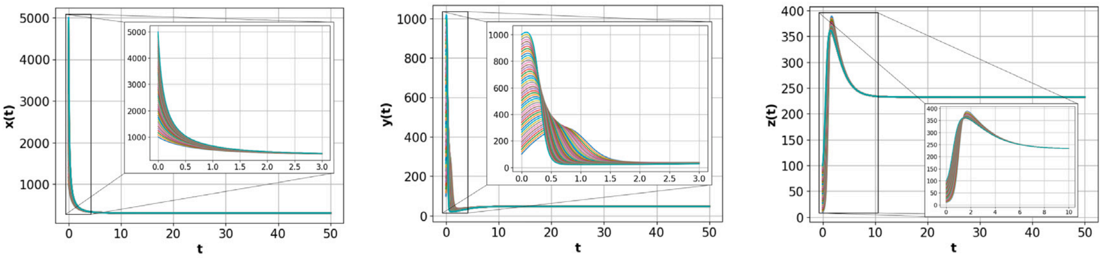
**Table 5**
Summarize results table in the variations of $b$.

| Case | Iteration | Cycle | Time/iter | Total time | Data type | MSE | RMSE | MAE |
|---|---|---|---|---|---|---|---|---|
| I | 960 | 30 | 2.71 s | 2597.407 s | Training | 2.607e−2 | 1.607e−1 | 2.231e−1 |
| | | | | | Validation | 2.279e−2 | 1.509e−1 | 2.014e−1 |
| | | | | | Testing | 2.279e−2 | 1.510e−1 | 2.013e−1 |
| II | 1212 | 30 | 2.74 s | 3324.189 s | Training | 9.519e−3 | 9.753e−2 | 8.695e−2 |
| | | | | | Validation | 9.475e−3 | 9.734e−2 | 8.820e−2 |
| | | | | | Testing | 9.480e−3 | 9.731e−2 | 8.819e−2 |
| III | 1131 | 30 | 2.77 s | 3132.804 s | Training | 2.888e−2 | 1.697e−1 | 1.695e−1 |
| | | | | | Validation | 2.861e−2 | 1.691e−1 | 1.532e−1 |
| | | | | | Testing | 2.860e−2 | 1.690e−1 | 1.520e−1 |
| IV | 1260 | 30 | 2.66 s | 3356.458 s | Training | 1.052e−2 | 1.025e−1 | 1.068e−1 |
| | | | | | Validation | 1.039e−2 | 1.019e−1 | 1.051e−1 |
| | | | | | Testing | 1.041e−2 | 1.020e−1 | 1.060e−1 |
| IV | 1064 | 30 | 2.63 s | 2796.618 s | Training | 3.843e−2 | 1.953e−1 | 2.14e−1 |
| | | | | | Validation | 3.646e−2 | 1.861e−1 | 1.862e−1 |
| | | | | | Testing | 3.651e−2 | 1.861e−1 | 1.863e−1 |

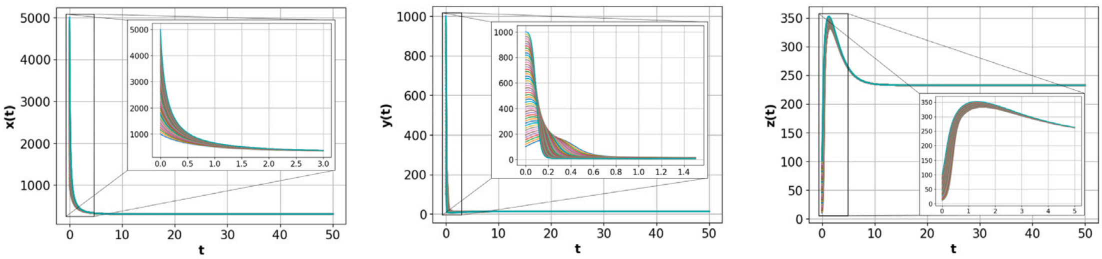
suggest that r-PINN-Adam is a promising approach for solving the problem at hand and may have broader implications for similar applications in the field.



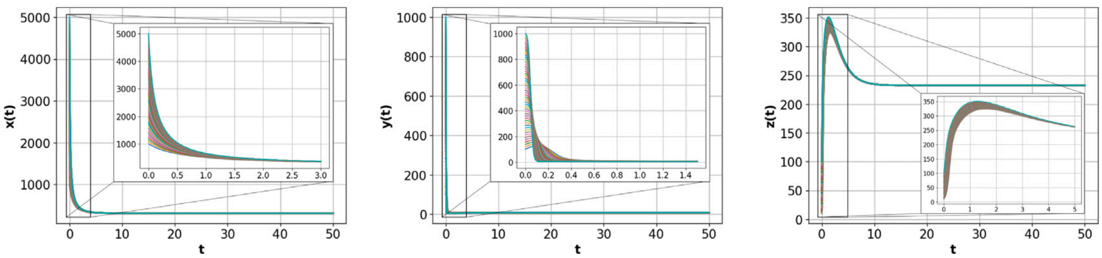

**(a)** Case 1: $b = 0.005$



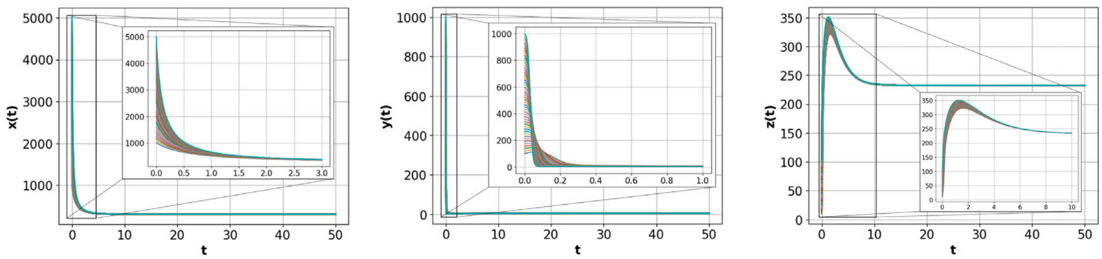

**(b)** Case 2: $b = 0.015$




**(c)** Case 3: $b = 0.05$

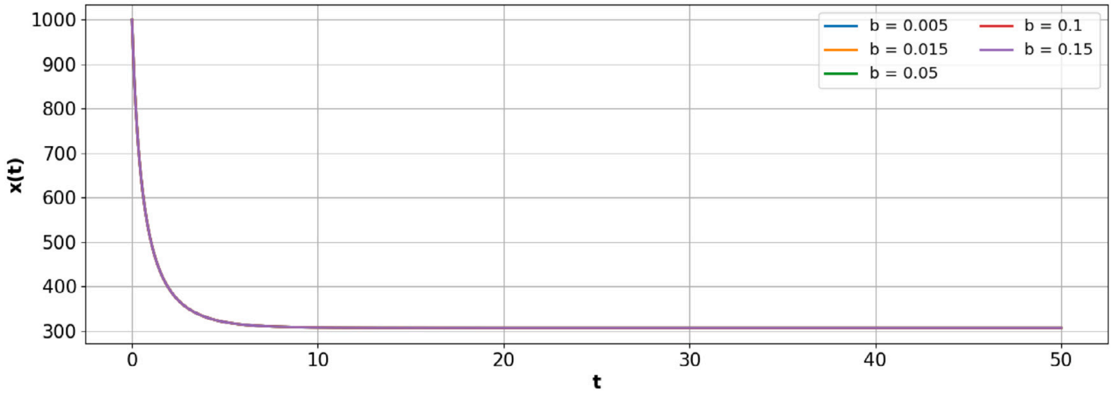



**(d)** Case 4: $b = 0.1$

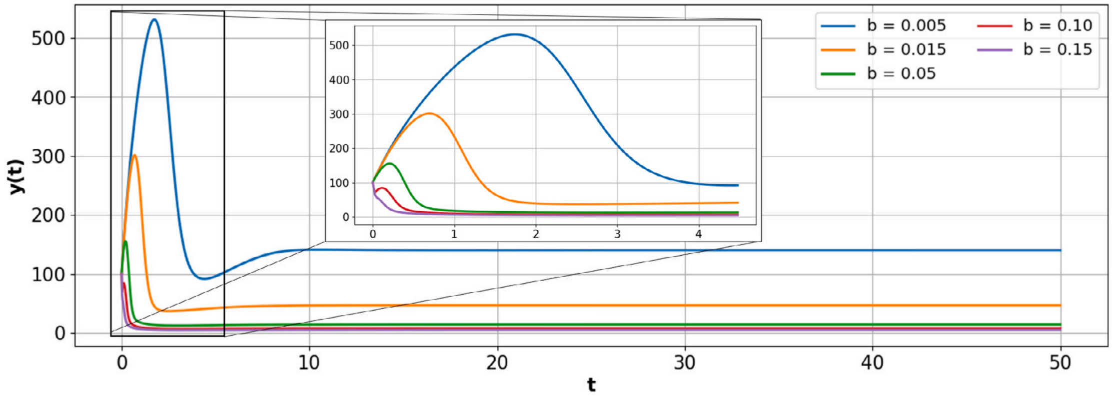



**(e)** Case 5: $b = 0.15$

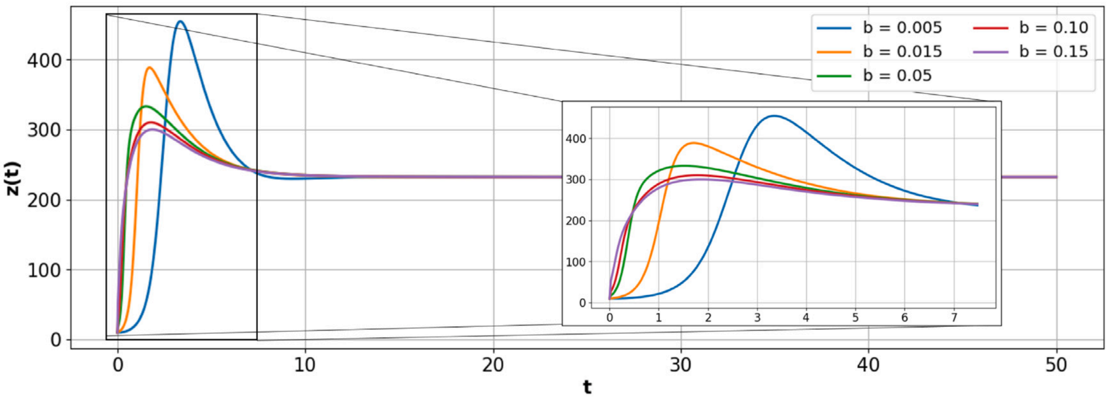
**Fig. 19.** The trajectory of $x(0)$ (left), $y(0)$ (mid), and $z(0)$ (right) for all case variations of $b$.




**(a)** $x(t)$



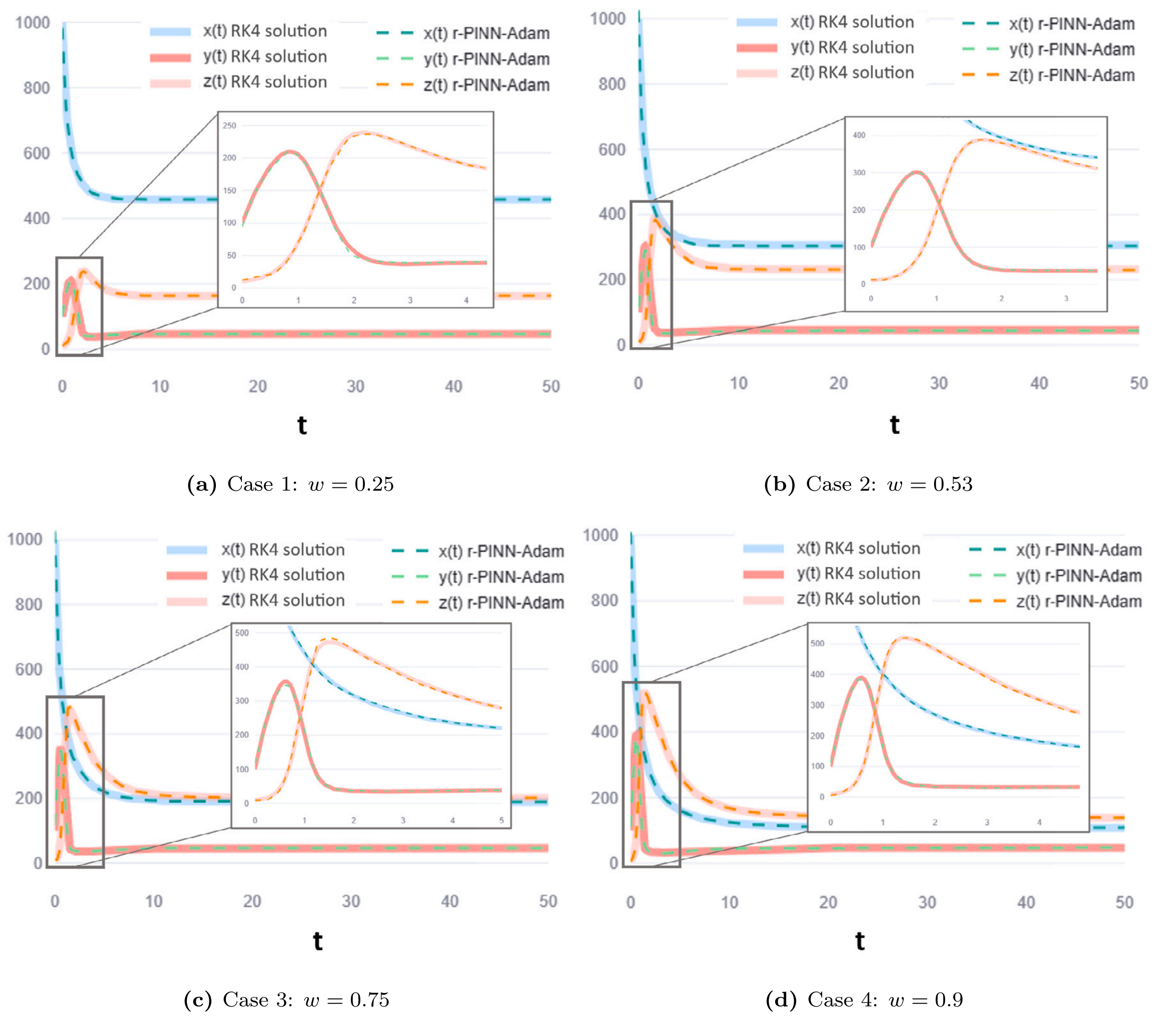

**(b)** $y(t)$




**(c)** $z(t)$

**Fig. 20.** Approximate solutions of $x(t), y(t)$ and $z(t)$ using r-PINN-Adam algorithm with the variation values of $b$ on population dynamic system of sea turtles.

Here are some recommendations for future work to enhance the performance of the r-PINN-Adam algorithm in solving DE problems. As the accuracy of the ANN model improves, there is a need to increase its architectural complexity, which results in a large number of weight parameters that can affect the training performance. Therefore, weight parameter compression is crucial for reducing model storage while maintaining accuracy. The low-rank matrix factorization method could also be employed in future research. By decomposing $\theta$ into two or more smaller matrices, this technique may accelerate the training process.




(a) Case 1: $w = 0.25$

(b) Case 2: $w = 0.53$

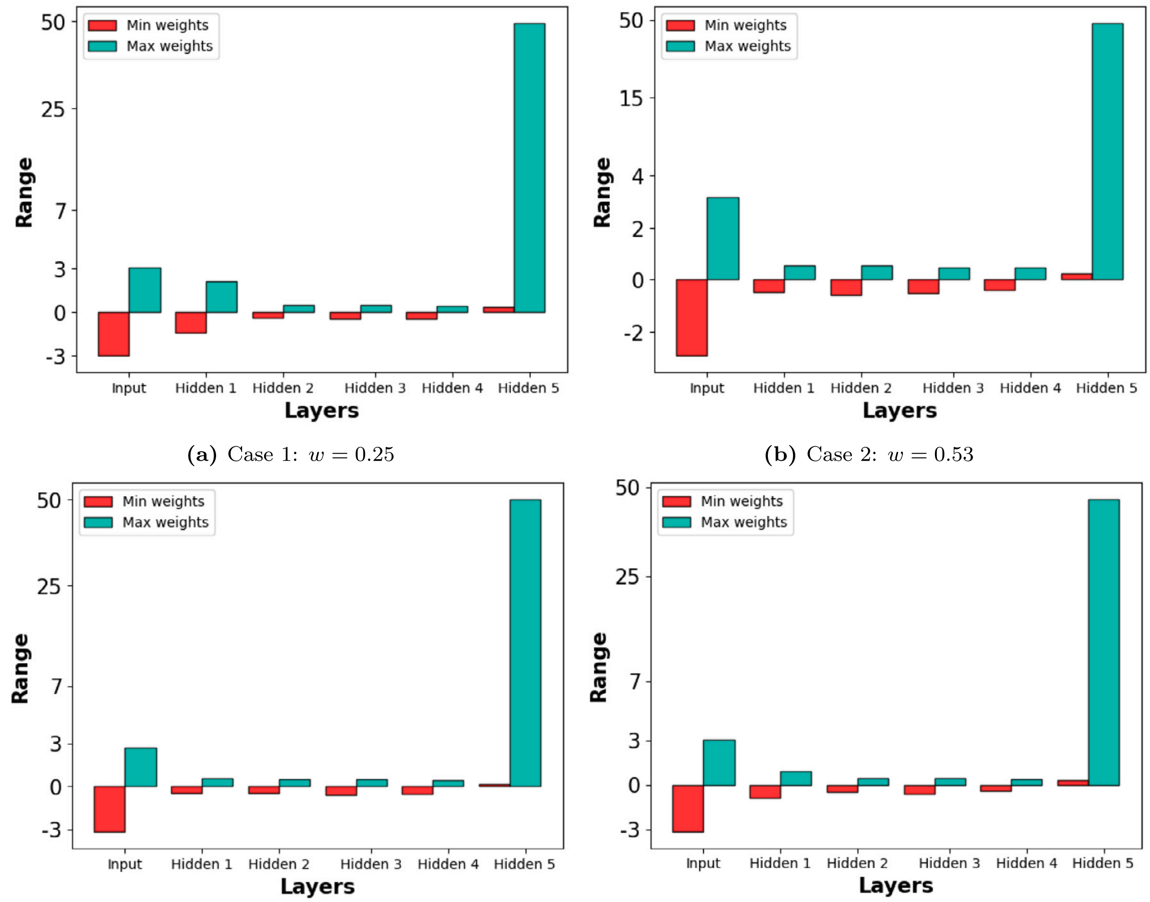
(c) Case 3: $w = 0.75$

(d) Case 4: $w = 0.9$

**Fig. 21.** Comparison of RK4 solutions and approximate solutions using r-PINN-Adam for all case variations of $w$.

**Table 6**
Results summaryof r-PINN-Adam performance in the variations of $w$.

| Case | Iteration | Cycle | Time/iter | Total time | Data type | MSE | RMSE | MAE |
|---|---|---|---|---|---|---|---|---|
| I | 1233 | 30 | 2.70 s | 3333.570 s | Training | 8.882e−3 | 9.415e−2 | 7.811e−2 |
| | | | | | Validation | 8.788e−3 | 9.374e−2 | 7.384e−2 |
| | | | | | Testing | 8.738e−3 | 9.347e−2 | 7.384e−2 |
| II | 1212 | 30 | 2.71 s | 3335.617 s | Training | 2.349e−3 | 4.847e−2 | 3.462e−2 |
| | | | | | Validation | 2.370e−3 | 4.868e−2 | 3.764e−2 |
| | | | | | Testing | 2.372e−3 | 4.868e−2 | 3.763e−2 |
| III | 1131 | 30 | 2.76 s | 3020.667 s | Training | 4.066e−3 | 6.361e−2 | 6.481e−2 |
| | | | | | Validation | 3.870e−3 | 6.237e−2 | 6.340e−2 |
| | | | | | Testing | 3.891e−3 | 6.238e−2 | 6.341e−2 |
| IV | 979 | 30 | 2.52 s | 2470.895 s | Training | 1.707e−2 | 1.305e−1 | 1.315e−1 |
| | | | | | Validation | 1.701e−2 | 1.292e−1 | 1.360e−1 |
| | | | | | Testing | 1.669e−2 | 1.291e−1 | 1.358e−1 |




(a) Case 1: $w = 0.25$

(b) Case 2: $w = 0.53$

(c) Case 3: $w = 0.75$

(d) Case 4: $w = 0.9$

**Fig. 22.** The best weights and biases range distribution in each layer for all case variations of $w$.

Indeed, the activation function in the ANN plays a important role in influencing accuracy and effectiveness. Investigating the utilization of mathematical functions such as Fourier, Euler, or other new activation functions could potentially enhance the performance of the r-PINN-Adam training process. Lastly, in the future, this approach could be applicable to solve another DE problems, including real-world applications of PDE.

## CRediT authorship contribution statement

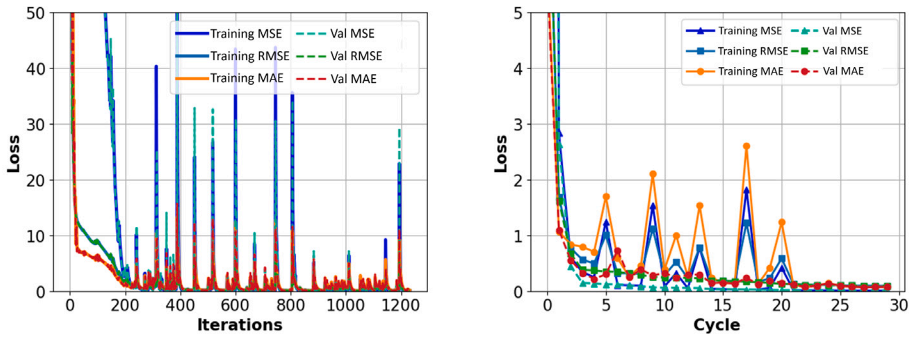
**Danang A. Pratama:** Conceptualization, Methodology, Software, Writing – original draft. **Maharani A. Bakar:** Conceptualization, Validation, Supervision, Project administration. **Ummu Atiqah Mohd Roslan:** Resources, Review & editing. **Sugiyarto Surono:** Resources, Writing – review & editing. **A. Salhi:** Formal analysis, Review & editing, Funding acquisition.

## Declaration of competing interest

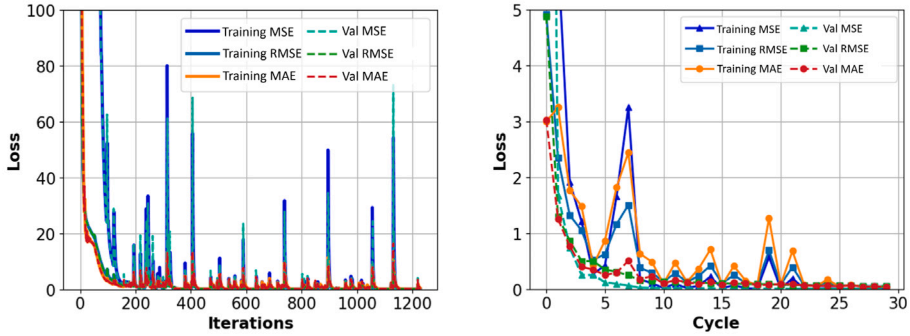
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

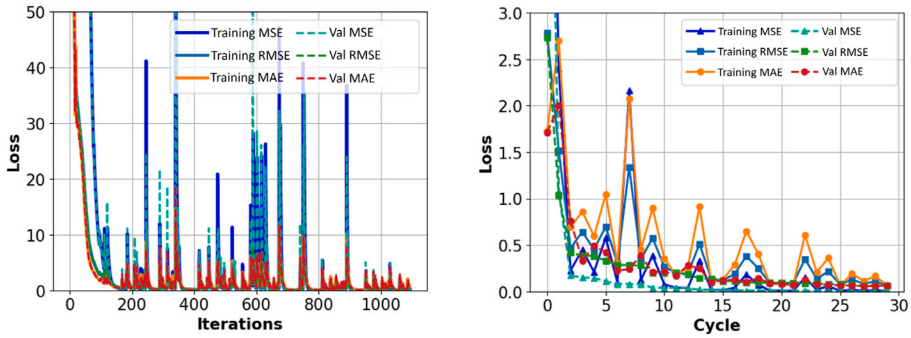
No data was used for the research described in the article.



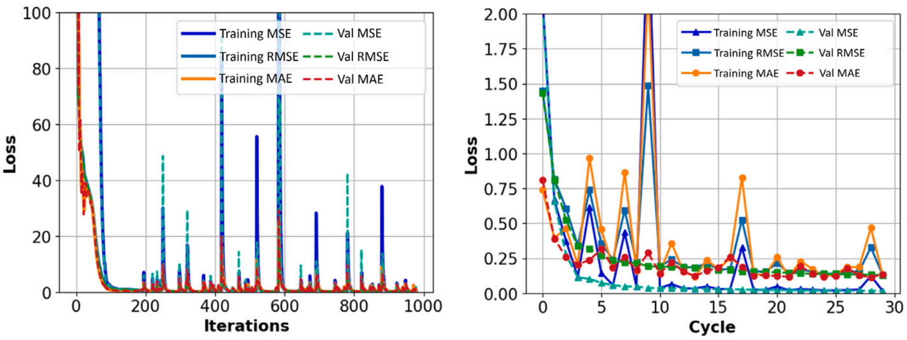

**(a)** Case 1: $w = 0.25$




**(b)** Case 2: $w = 0.53$

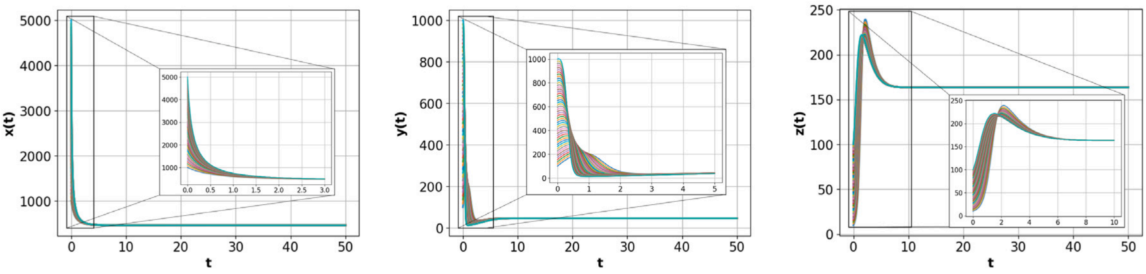



**(c)** Case 3: $w = 0.75$

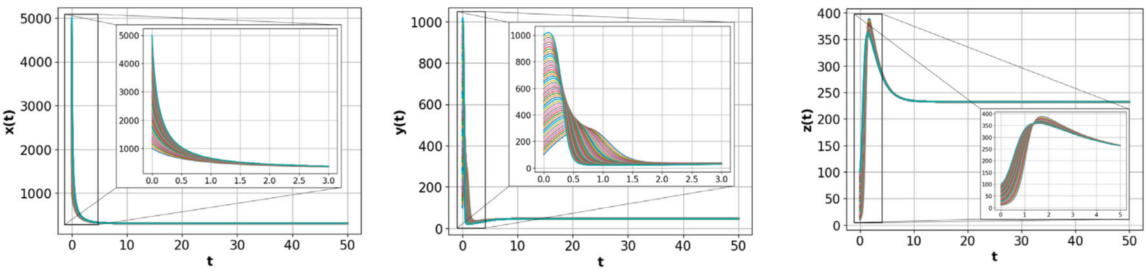



**(d)** Case 4: $w = 0.9$

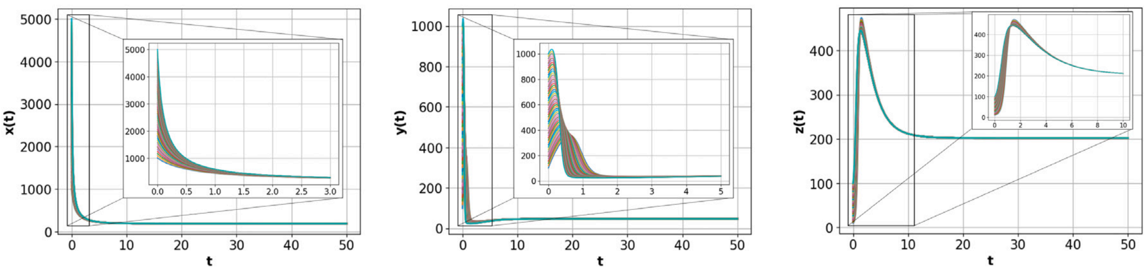
**Fig. 23.** Performance of r-PINN-Adam: loss function (left) and the minimum loss function for each cycle (right) for all case variations of $w$.



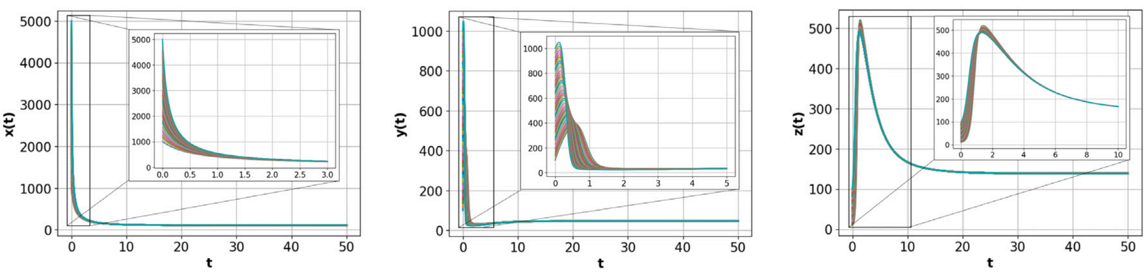

**(a)** Case 1: $w = 0.25$




**(b)** Case 2: $w = 0.53$

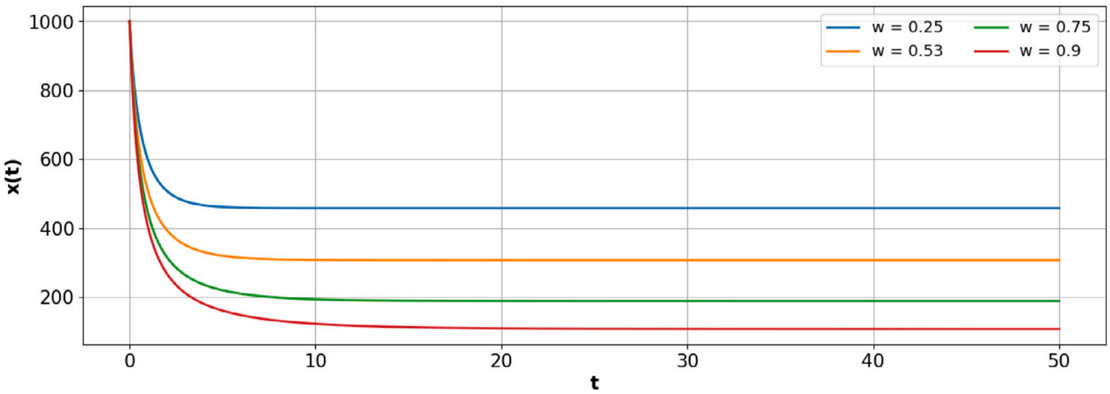



**(c)** Case 3: $w = 0.75$

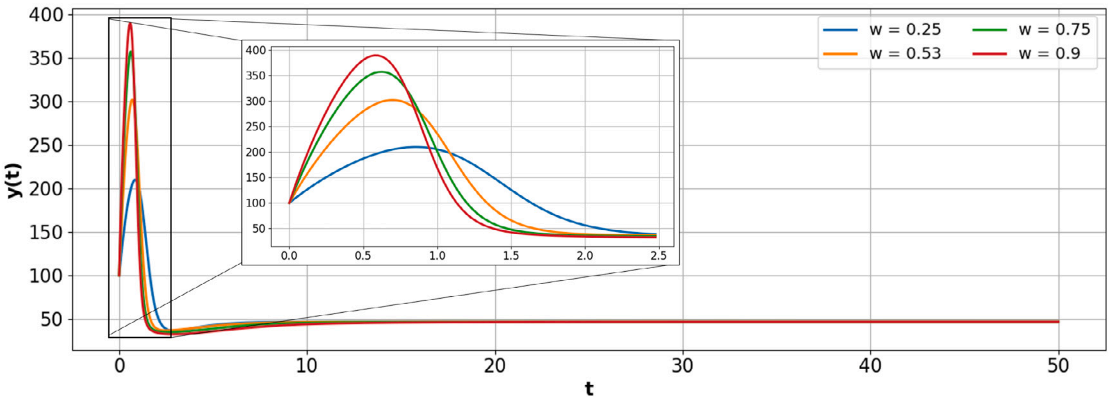



**(d)** Case 4: $w = 0.9$

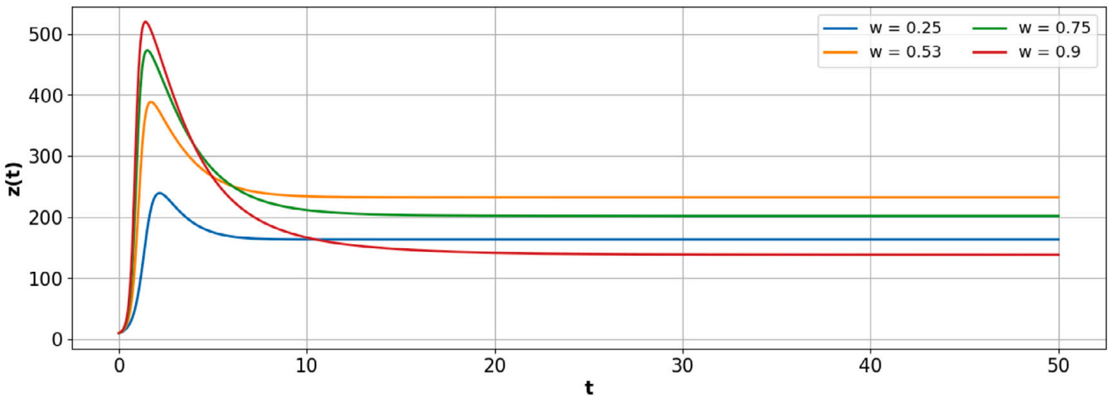
**Fig. 24.** The trajectory of $x(0)$ (left), $y(0)$ (mid), and $z(0)$ (right) for all case variations of $w$.




**(a)** $x(t)$




**(b)** $y(t)$




**(c)** $z(t)$

**Fig. 25.** Approximate solutions of $x(t)$, $y(t)$ and $z(t)$ obtained by the r-PINN-Adam algorithm under the influence of variations of $w$ on population dynamic system of sea turtles.

## Acknowledgments

This project is funded by Ministry of Higher Education (MOHE), Malaysia under the Fundamental Research Grant Scheme (FRGS), Ref: FRGS/1/2022/STG06/UMT/03/3. The authors are grateful to ESRC, United Kingdom (Grant ES/L011859/1) for partially funding this research.

## References


[1] Lei J, Booth DT. Who are the important predators of sea turtle nests at wreck rock beach? PeerJ 2017;5:e3515.
[2] Heithaus MR. 10 Predators, prey, and the ecological roles of sea turtles. Biol Sea Turtles, Volume III 2013;3:249.
[3] Crouse DT, Crowder LB, Caswell H. A stage-based population model for loggerhead sea turtles and implications for conservation. Ecology 1987;68(5):1412–23.
[4] Crowder LB, Crouse DT, Heppell SS, Martin TH. Predicting the impact of turtle excluder devices on loggerhead sea turtle populations. Ecol Appl 1994;4(3):437–45.
[5] Liu X, Chen L. Complex dynamics of holling type II Lotka–Volterra predator–prey system with impulsive perturbations on the predator. Chaos Solitons Fractals 2003;16(2):311–20.
[6] Khairuddin MAA, Roslan UAM, Mohd MH, Rusli MU. Sea turtle nest conservation-exploitation prey-predator model: Case study of terengganu. In: AIP conference proceedings, vol. 2355, no. 1. AIP Publishing LLC; 2021, 040001.
[7] Wei Y, Song B, Yuan S. Dynamics of a ratio-dependent population model for green sea turtle with age structure. J Theoret Biol 2021;516:110614.
[8] Bajalan S, Bajalan N. Novel ANN method for solving ordinary and time-fractional black–scholes equation. Complexity 2021;2021:1–15.
[9] Ureña F, García Á, Vargas AM. Preface to "applications of partial differential equations in engineering". Mathematics 2022;11(1):199.
[10] Laham MF, Krishnarajah I, Jumaat AK. A numerical study on predator prey model. In: International journal of modern physics: conference series, vol. 9. World Scientific; 2012, p. 347–53.
[11] Paul S, Mondal SP, Bhattacharya P. Numerical solution of Lotka Volterra prey predator model by using runge–kutta–fehlberg method and Laplace adomian decomposition method. Alex Eng J 2016;55(1):613–7.
[12] Du M, Ning P, Wang Y. Numerical solution of a class of predator-prey systems with complex dynamics characters based on a sinc function interpolation collocation method. Complexity 2020;2020:1–34.
[13] Berg J, Nyström K. A unified deep artificial neural network approach to partial differential equations in complex geometries. Neurocomputing 2018;317:28–41.
[14] Nüsken N, Richter L. Solving high-dimensional hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. Partial Differ Equ Appl 2021;2:1–48.
[15] Maria Antony AN, Narisetti N, Gladilin E. FDM data driven U-net as a 2D Laplace PINN solver. Sci Rep 2023;13(1):9116.
[16] Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans Neural Netw 1998;9(5):987–1000.
[17] Malek A, Beidokhti RS. Numerical solution for high order differential equations using a hybrid neural network-optimization method. Appl Math Comput 2006;183(1):260–71.
[18] Sabir Z, Ali MR, Sadat R. Gudermannian neural networks using the optimization procedures of genetic algorithm and active set approach for the three-species food chain nonlinear model. J Ambient Intell Humaniz Comput 2022;1–10.
[19] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[20] Jagtap AD, Kawaguchi K, Karniadakis GE. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. J Comput Phys 2020;404:109136.
[21] Jagtap AD, Kawaguchi K, Em Karniadakis G. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. Proc R Soc Lond Ser A Math Phys Eng Sci 2020;476(2239):20200334.
[22] McClenny L, Braga-Neto U. Self-adaptive physics-informed neural networks using a soft attention mechanism. 2020, arXiv preprint arXiv:2009.04544.
[23] Pratama DA, Abo-Alsabeh RR, Bakar MA, Salhi A, Ibrahim NF. Solving partial differential equations with hybridized physic-informed neural network and optimization approach: Incorporating genetic algorithms and L-BFGS for improved accuracy. Alex Eng J 2023;77:205–26.
[24] Pratama DA, Bakar MA, Ibrahim NF, Idris R, Mohamed N. Physical restriction neural networks with restarting strategy for solving mathematical model of thermal heat equation for early diagnose breast cancer. Results Appl Math 2023;19:100384.
[25] Bjorndal KA, Bolten AB, Chaloupka MY. Green turtle somatic growth model: evidence for density dependence. Ecol Appl 2000;10(1):269–82.
[26] Wyneken J, Burke T, Salmon M, Pedersen D. Egg failure in natural and relocated sea turtle nests. J Herpetol 1988;88–96.
[27] Abella E, García-Cerdá RM, Marco A. Estimating the fertilization rate of sea turtle nests: comparison of two techniques. Basic Appl Herpetol 2017;31:33–44.
[28] IAC. An introduction to sea turtles of the world. 2004.
[29] Gyuris E. The rate of predation by fishes on hatchlings of the green turtle (Chelonia mydas). Coral Reefs 1994;13:137–44.
[30] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 1026–34.
[31] Panghal S, Kumar M. Optimization free neural network approach for solving ordinary and partial differential equations. Eng Comput 2020;1–14.
[32] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. In: 12th {USeNIX} symposium on operating systems design and implementation ({oSDI} 16). 2016, p. 265–83.
[33] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: 3rd International conference on learning representations. 2015, p. 15.
[34] Maharani M, A. S. Restarting from specific points to cure breakdown in lanczos-type algorithms. J Math Fundamental Sci 2015;47(2):167–84.