Discrete Dynamics in Nature and Society

Data-driven Dynamics Modeling and Analysis Using Computation Intelligence

Lead Guest Editor: Bin Xu
Guest Editors: Chenguang Yang and Hai Wang

Data-driven Dynamics Modeling and Analysis Using Computation Intelligence

Discrete Dynamics in Nature and Society

Data-driven Dynamics Modeling and Analysis Using Computation Intelligence

Lead Guest Editor: Bin Xu
Guest Editors: Chenguang Yang and Hai Wang

Contents

Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 1241761, 44 pages
https://doi.org/10.1155/2022/1241761

Research Article

Mathematical Analysis of the Prey-Predator System with Immigrant Prey Using the Soft Computing Technique

Naveed Ahmad Khan,[1] **Muhammad Sulaiman**,[1] **Jamel Seidu**,[2] **and Fahad Sameer Alshammari**[3]

[1]*Department of Mathematics, Abdul Wali Khan University Mardan, Mardan, KP, Pakistan*
[2]*School of Railways and Infrastructure Development, University of Mines and Technology (UMaT) Essikado, Sekondi-Takoradi, Ghana*
[3]*Department of Mathematics, College of Science and Humanities in Alkharj, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia*

Correspondence should be addressed to Jamel Seidu; jseidu@umat.edu.gh

Received 28 May 2022; Revised 20 August 2022; Accepted 19 September 2022; Published 13 October 2022

Academic Editor: Chenguang Yang

In this paper, a mathematical model for the system of prey-predator with immigrant prey has been analyzed to find an approximate solution for immigrant prey population density, local prey population density, and predator population density. Furthermore, we present a novel soft computing technique named LeNN-WOA-NM algorithm for solving the mathematical model of the prey-predator system with immigrant prey. The proposed algorithm uses a function approximating ability of Legendre polynomials based on Legendre neural networks (LeNNs), global search ability of the whale optimization algorithm (WOA), and a local search mechanism of the Nelder–Mead algorithm. The LeNN-WOA-NM algorithm is applied to study the effect of variations on the growth rate, the force of interaction, and the catching rate of local prey and immigrant prey. The statistical data obtained by the proposed technique establish the effectiveness of the proposed algorithm when compared with techniques in the latest literature. The efficiency of solutions obtained by LeNN-WOA-NM is validated through performance measures including absolute errors, MAD, TIC, and ENSE.

1. Introduction

A prey-predator system is one of the prevalent phenomena in nature. The interaction between predators and prey in any environment was first introduced by Lotka and Volterra in 1926 [1]. Holling in 1966 [2] elaborated prey-predictor models with different kinds of functional responses for predation. All these models are inspired by biological phenomena and presented by nonlinear ordinary and partial differential equations. Danca et al. [3] study the detailed analysis of a nonlinear prey-predator model. A connection between predators and prey has a long history and will continue as a governing theme in biomathematics because of its universal significance [4]. Researchers have made many changes by introducing different facets to the predator-prey system such as delay in predator growth, harvesting prey and predators, and providing additional food to a predator for sustaining prey population and prey diseases [5–7]. Cai et al. [8] studied the dynamical system of prey-predator with Allee effects in prey growth. A stage-structured predator-prey model with gestation delay is investigated by [9]. In [10, 11], bifurcation, chaos and dynamic behavior of the nonlinear discrete-time predator-prey system is studied. Huang et al. [12] study the stability analysis of the model with consideration of the prey refuge. Studies by Din [13], Weide et al. [14], and Gong et al. [15, 16] are presented for the discrete-time nonlinear prey-predator types of model. Hadeler and Freedman [17] provide extensive references to real-world examples of three-species ecoepidemiological systems of sound prey, infected prey, and predators.

In recent years, there has been a substantial increase in the amount of attention paid to population biology by scientists due to the significant applications it has in ecology. It bridges the gap between mathematics and biology. The dynamical systems in biology have been investigated to interpret various problems. The Lotka–Volterra population model is a well-known mathematical model that describes biological systems [18]. In real life, there is a strong correlation between size, age, and developmental stages of different populations of species. It is an important strategy to incorporate all these variables in the mathematical modeling of populations of different spices. To develop more realistic and accurate mathematical models, many scientists have suggested noise-induced models [19, 20] and spatial models [21]. In biomathematical problems, researchers utilize mathematical modeling and simulations along with biological structures to describe the phenomena by the system of nonlinear differential equations. These nonlinear models are considered stiff and unrealistic, and therefore, finding exact and semianalytical solutions for such problems is challenging because of nonlinearity. Analytical approaches for solving nonlinear problems are mostly based on Laplace or Fourier transformation, Laguerre's integral formula, and the Grunwald–Letnikov concept [22, 23]. When tackling complex problems, these techniques may be challenging to use; in addition, the solution is provided in a closed form that necessitates the evaluation of special functions using complex expressions, such as the Mittag-Leffler function [24].

In recent years, researchers have been working to develop new techniques for finding approximate solutions to nonlinear models; e.g., the Laplace Adomian decomposition method [25], the new coupled fractional reduced differential transform method [26], the Runge–Kutta–Fehlberg method [27], the finite element method [28], the Sumudu decomposition method [29], the implicit Adams methods [30], the confidence domain technique [31], and the homotopy analysis method [32] have become much more significant to get accurate solutions. All these deterministic approaches have their own advantages, applicability, and drawbacks. With great interest, it is noted that such techniques are gradient-based and call for information about the problem beforehand. The availability of several local optima, which leads to solutions where global optimality cannot be easily ensured, is one of the fundamental limitations of gradient-based approaches. Global optimality is sought in gradient-based approaches by randomly scanning the design space from various starting points. However, this causes the technique to become sluggish and computationally inefficient for complex nonlinear optimization problems [33, 34]. In addition, the Runge–Kutta methods are self-starting and stable techniques that can be easily implemented to calculate the solution to different problems. The main drawbacks of the Runge–Kutta methods are that they take longer time to calculate solutions than other multistep methods with equivalent precision, and it is difficult for them to get accurate global estimates of the truncation error [35]. To overcome these drawbacks, a stochastic metaheuristic approach with artificial neural networks is developed, which is free of a gradient and does not require any prior information about the problem. The majority of metaheuristic techniques are inspired by natural, physical, or biological processes and make use of a variety of operators to mimic the fundamental behavior. The harmony between exploration and exploitation is a recurring subject in all metaheuristics.

In recent times, artificial neural network (ANN)-based stochastic algorithms with global and local search optimizers have been designed to solve differential equations representing physical phenomena including flow in a circular cylindrical conduit via electrohydrodynamics [36], a model of an immobilized enzyme system that follows the Michaelis–Menten (MM) kinetics for a microdisk biosensor [37], flow of Johnson–Segalman fluid on the surface of an infinitely long vertical cylinder [38], and beam-column designs [39]. The abovementioned techniques motivate authors to design a new soft computing algorithm, the LeNN-WOA-NM algorithm, to find approximate series solutions using Legendre polynomials for the model presenting the prey-predator system with immigrant prey. The salient features of the paper are summarized as follows:

(i) A mathematical model for a prey-predator system with immigrant prey is formulated and analyzed to study the influence of variations on the growth rate, force of interaction, and the catching rate of local and immigrant prey.

(ii) Artificial neural network-based weighted Legendre polynomials are used to construct the model of approximate solutions for the prey-predator model. A fitness function based on mean square errors is designed to assess unknown parameters with the help of global search ability of whale optimization and a local search mechanism of the Nelder–Mead algorithm.

(iii) The suggested technique can result in adequate solutions for nonlinear hard problems for which no exact algorithm exists that can solve them in a reasonable amount of time.

(iv) Performance of the proposed algorithm is validated in terms of absolute errors, mean absolute deviation, Theil's inequality coefficient, Nash–Sutcliffe efficiency, and error in Nash–Sutcliffe efficiency.

(v) Approximate solutions, convergence of fitness, and performance measures obtained by the LeNN-WOA-NM algorithm for prey-predator systems with immigrant prey are shown through different graphs and tables, which shows the dominance and robustness of the proposed algorithm in solving real-world problems.

(vi) Unlike most classical methods, the LeNN-WOA-NM algorithm requires no gradient information and therefore can be used with nonanalytic, black-box, or simulation-based objective functions to approximate complex problems.

2. Problem Formulation

Goteti developed the mathematical model for a prey-predator system with immigrant prey to understand the interaction and communication between predators and immigrant prey. The model is assumed to follow mass action theory and consists of prey population density, which is defined as follows:

$$N(t) = X(t) + S(t), \tag{1}$$

where $S(t)$ and $X(t)$ denote the population density of local and immigrant prey, respectively. Population density of the predator is denoted by $Y(t)$.

The following assumptions are used in formulating the mathematical model for the prey-predator system with immigrant prey:

(i) In the presence of a predator, the population of prey is classified into two subcategories named local prey $S(t)$ and immigrant prey $X(t)$.

(ii) In absence of the predator, the population growth of local prey logistically increases with an intrinsic growth rate α_1 with environmental carrying capacity denoted by k_1.

(iii) With availability of the local and immigrant prey population, the population of the predator grows logistically with growth rates c_1 and c_2, while suffering loss of populations is denoted by μ_1 and μ_2.

(iv) Immigrant and local prey can reproduce, and therefore, it is assumed that birth rates should be positive. The growth rate of immigrant prey increases at the rate α_2 with environmental carrying capacity denoted by k_2.

(v) It is assumed that immigrant prey is a natural choice of predators. β_1 and β_2 denote the positive and negative force of interaction between local and immigrant prey.

(vi) It is assumed that the local prey and immigrant prey are caught by the predator at the rate of γ_1 and γ_2, respectively.

A mathematical model for a prey-predator system with immigrant prey is given by the following system of differential equations:

$$\begin{aligned} &\frac{dS}{dt} - S\left(\alpha_1 - \frac{\alpha_1 S}{k_1}\right) - \beta_1 SX + \gamma_1 SY = 0, \\ &\frac{dX}{dt} - X\left(\alpha_2 - \frac{\alpha_2 X}{k_2}\right) + \beta_2 SX + \gamma_2 XY = 0, \end{aligned} \tag{2}$$

$$\frac{dY}{dt} - c_1 SY - c_2 XY + \mu_1 Y + \mu_2 Y^2 = 0, \tag{3}$$

with initial populations

$$S = S_0, X = X_0, Y = Y_0 \quad at \quad t = 0. \tag{4}$$

3. Approximate Solutions and Weighted Legendre Polynomials

The Legendre polynomials are denoted by $L_n(t)$, where n denotes the order of Legendre polynomials. These polynomials constitute the set of orthogonal polynomials on $[-1, 1]$. The first eleven Legendre polynomials are given in Table 1. High-order Legendre polynomials are generated by the following recursive formula:

$$L_{n+1}(r) = \frac{1}{n+1}\left[(2n+1) r L_n(r) - n L_{n-1}(r)\right]. \tag{5}$$

We consider an approximate series solution for equations (2)–(4) representing the prey-predator model with immigrant prey as follows:

$$\begin{cases} S_{\text{approx}}(t) = \sum_{n=1}^{34} \zeta_n L_n(\psi_n(t) + \theta_n), \\ X_{\text{approx}}(t) = \sum_{n=35}^{68} \zeta_n L_n(\psi_n(t) + \theta_n), \\ Y_{\text{approx}}(t) = \sum_{n=69}^{102} \zeta_n L_n(\psi_n(t) + \theta_n), \end{cases} \tag{6}$$

where ζ_n, ψ_n, and θ_n are unknown parameters.

Since nth order continuous derivatives of system (6) exist, we consider the first derivative of system (6) as follows:

$$\begin{cases} \frac{d}{dt}\left(S_{\text{approx}}\right) = \sum_{n=1}^{34} \zeta_n L_n'(\psi_n(t) + \theta_n), \\ \frac{d}{dt}\left(X_{\text{approx}}\right) = \sum_{n=35}^{68} \zeta_n L_n'(\psi_n(t) + \theta_n), \\ \frac{d}{dt}\left(Y_{\text{approx}}\right) = \sum_{n=69}^{102} \zeta_n L_n'(\psi_n(t) + \theta_n), \end{cases} \tag{7}$$

and we plug equations (6)–(7) in governing ordinary differential equations. Equations (2)–(4) will be transformed into an equivalent algebraic system of equations that can be solved for unknown parameters ζ_n, ψ_n, and θ_n using the LeNN-WOA-NM algorithm.

4. Fitness Function Formulation

The mean square error (MSE)-based fitness function for solving the prey-predator model for equations (2)-(4) can be written as follows:

$$\text{Minimize}\, \varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6, \tag{8}$$

where ϵ_1 to ϵ_6 are the fitness-based errors for equations (2)–(4) along with the initial conditions and are given as follows:

Table 1: Eleven Legendre polynomials with an independent variable t.

n	$L_n(t)$
0	1
1	t
2	$1/2(3t^2-1)$
3	$1/2(5t^3-3t)$
4	$1/8(35t^4-30t^2+3)$
5	$1/8(63t^5-70t^3+15t)$
6	$1/16(231t^6-315t^4+105t^2-5)$
7	$1/16(429t^7-693t^5+315t^3-35t)$
8	$1/128(6435t^8-12012t^6+6930t^4-1260t^2+35)$
9	$1/128(12155t^9-25740t^7+18018t^5-4620t^3+315t)$
10	$1/256(46189t^{10}-109395t^8+90090t^6-30030t^4+3465t^2-63)$

$$\begin{cases} \varepsilon_1 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{34}\left(\dfrac{dS_n}{dt} - S_n\left(\alpha_1 - \dfrac{\alpha_1 S_n}{k_1}\right) - \beta_1 S_n X_n + \gamma_1 S_n Y_n\right)^2, \\ \varepsilon_2 = \dfrac{1}{N}\displaystyle\sum_{n=35}^{68}\left(\dfrac{dX_n}{dt} - X_n\left(\alpha_2 - \dfrac{\alpha_2 X_n}{k_2}\right) + \beta_2 S_n X_n + \gamma_2 X_n Y_n\right)^2, \\ \varepsilon_3 = \dfrac{1}{N}\displaystyle\sum_{n=69}^{102}\left(\dfrac{dY_n}{dt} - c_1 S_n Y_n - c_2 X_n Y_n + \mu_1 Y_n + \mu_2 Y_n^2\right)^2, \\ \varepsilon_4 = (S_0 - 0.5)^2, \\ \varepsilon_5 = (X_0 - 0.5)^2, \\ \varepsilon_6 = (Y_0 - 0.5)^2. \end{cases} \tag{9}$$

5. Optimization Network

5.1. Whale Optimization Algorithm. The whale optimization algorithm is a nature-inspired metaheuristic algorithm designed by Mirjalili and Lewis [40]. The working strategy of WOA is inspired by foraging behavior of humpback whales. The humpback whales chase prey or krill by swimming around them in a molded way as shown in Figure 1. The mathematical model of each phase is explained below.

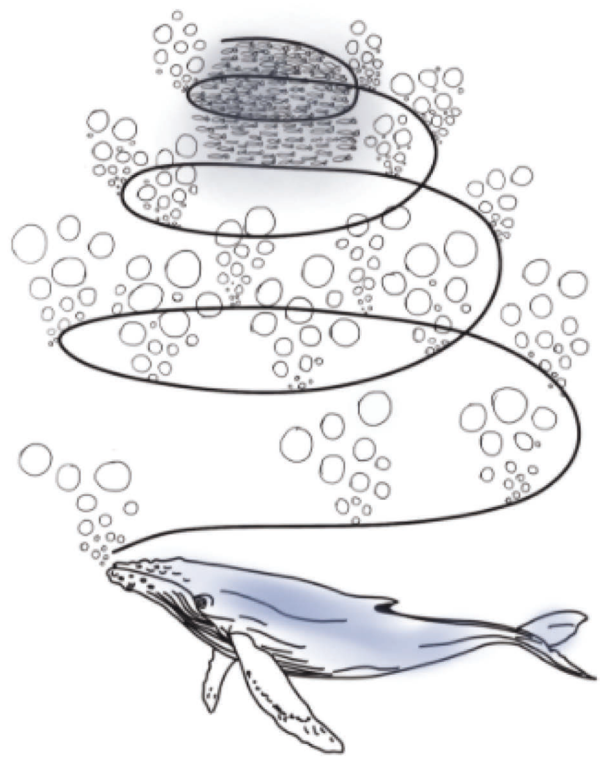


Figure 1: Unique bubble-net feeding method and spiral moment for updating the position of humpback whales.

5.1.1. Exploration Phase. Humpback whales encircle prey for hunting. Equations (10) and (11) mathematically model this behavior as follows:

$$\vec{E} = \left|\vec{C}\cdot\vec{Z}^*(t) - \vec{Z}(t)\right|, \tag{10}$$

$$\vec{Z}^*(t+1) = \vec{Z}^*(t) - \vec{A}\cdot D, \tag{11}$$

where t denotes the current iteration, Z^* has provided the best solution so far, and D gives the location of humpback whales to prey at each step. A and C are coefficient vectors which are defined as follows:

$$\vec{A} = 2\vec{a}\cdot\vec{r} - \vec{a}, \tag{12}$$

$$\vec{C} = 2\cdot\vec{r}. \tag{13}$$

where $r \in [0,1]$ is an arbitrary nominated vector. The value of a reduces from two to zero during exploration as well as in the exploitation phase. During helix-shaped movement, the distance between prey and the humpback whale is given as follows:

$$\vec{Z}(t+1) = D'\cdot e^{bl}\cdot\cos(2\pi l) + \vec{Z}^*(t), \tag{14}$$

where $D\prime$ represents the location of the ith whale to the prey and is defined as follows:

$$D\prime = \left|\vec{Z}^{*}(t) - \vec{Z}(t)\right|. \quad (15)$$

Furthermore, b denotes the state of the logarithmic helix and $l \in [-1, 1]$ is any arbitrary number. The shrinking surrounding technique of humpback whales during contracting loop is summarized as follows:

$$\vec{Z}(t+1) = \begin{cases} \vec{Z}(t) - \vec{A} \cdot D & \text{if } p \leq 0.5, \\ \vec{D}\prime \cdot e^{bl} \cdot \cos(2\pi l) + \vec{Z}^{*}(t) & \text{if } p \geq 0.5. \end{cases} \quad (16)$$

5.1.2. Exploitation Phase. In the exploration phase (searching for prey), the heterogeneity of the vector A is utilized. If $|A| > 1$, the position of the search agent is updated, and the entire mechanism is modeled by the following equations:

$$\vec{D} = \vec{C} \cdot \vec{Z}^{*}_{\text{rand}} - \vec{Z}, \quad (17)$$

$$\vec{Z}(t+1) = \vec{Z}_{\text{rand}} - \vec{A} \cdot \vec{D}, \quad (18)$$

where $\overrightarrow{Z_{\text{rand}}}$ is an arbitrary position vector taken from the current population.

Figure 2 shows the flowchart of the WOA. It can be seen that the WOA creates a random initial population from candidate space and evaluates it using an error-based fitness function when the optimization process starts. After finding the best solution, the algorithm repeatedly executes the following steps until ending criterion is achieved.

5.2. Nelder–Mead Algorithm. A Nelder–Mead (NM) algorithm is a direct search method also known as a downhill simplex method developed by Nelder and Mead in 1965 to solve different problems without any information about the gradient [41]. NM is a single path following a local search optimizer that can find good results if initialized with a better initial solution. A simplex consisting of $n + 1$ vertices is set up to minimize a function f with dimensions n [42]. The NM algorithm generates a sequence of simplices by following four basic procedures, namely, reflection, expansion, contraction, and shrink. Further details about the NM algorithm can be found in [43]. Figure 2 shows the working procedure of the NM algorithm.

In recent times, the most successful and effective trend in optimization is the action of integrating components from different methods. The foremost motivation behind the hybridization of diverse algorithmic ideas is to acquire better performing systems, which exploit and coalesce benefits of different techniques. Therefore, in this study, we have combined the global and local search optimization algorithms to achieve more efficient and robust solutions. The hybridized working procedure of the designed LeNN-WOA-NM algorithm is illustrated in Figure 2.

6. Performance Measures

To examine the performance of the proposed algorithm, performance indices are defined in terms of mean absolute deviation (MAD), Theil's inequality coefficient (TIC), and error in Nash–Sutcliffe efficiency (ENSE). Mathematical formulation for $S_{\text{approx}}(t)$, $X_{\text{approx}}(t)$, and $Y_{\text{approx}}(t)$ of the predator and prey model in the case of MAD, TIC, and ENSE is presented as follows:

$$[\text{MAD}_S, \text{MAD}_X, \text{MAD}_Y] = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n}\left|S(t_i) - S_{\text{approx}}(t_i)\right| \\ \frac{1}{n}\sum_{i=1}^{n}\left|X(t_i) - X_{\text{approx}}(t_i)\right| \\ \frac{1}{n}\sum_{i=1}^{n}\left|Y(t_i) - Y_{\text{approx}}(t_i)\right| \end{bmatrix}^{T}, \quad (19)$$

$$[TIC_S, TIC_X, TIC_Y] = \begin{bmatrix} \dfrac{\sqrt{(1/n)\sum_{i=1}^{n}\left(S(t_i) - S_{\text{approx}}(t_i)\right)^2}}{\sqrt{(1/n)\sum_{i=1}^{n}\left(S(t_i)\right)^2} + \sqrt{(1/n)\sum_{i=1}^{n}\left(S_{\text{approx}}(t_i)\right)^2}} \\ \dfrac{\sqrt{(1/n)\sum_{i=1}^{n}\left(X(t_i) - X_{\text{approx}}(t_i)\right)^2}}{\sqrt{(1/n)\sum_{i=1}^{n}\left(X(t_i)\right)^2} + \sqrt{(1/n)\sum_{i=1}^{n}\left(X_{\text{approx}}(t_i)\right)^2}} \\ \dfrac{\sqrt{(1/n)\sum_{i=1}^{n}\left(Y(t_i) - Y_{\text{approx}}(t_i)\right)^2}}{\sqrt{(1/n)\sum_{i=1}^{n}\left(Y(t_i)\right)^2} + \sqrt{(1/n)\sum_{i=1}^{n}\left(Y_{\text{approx}}(t_i)\right)^2}} \end{bmatrix}^{T}, \quad (20)$$

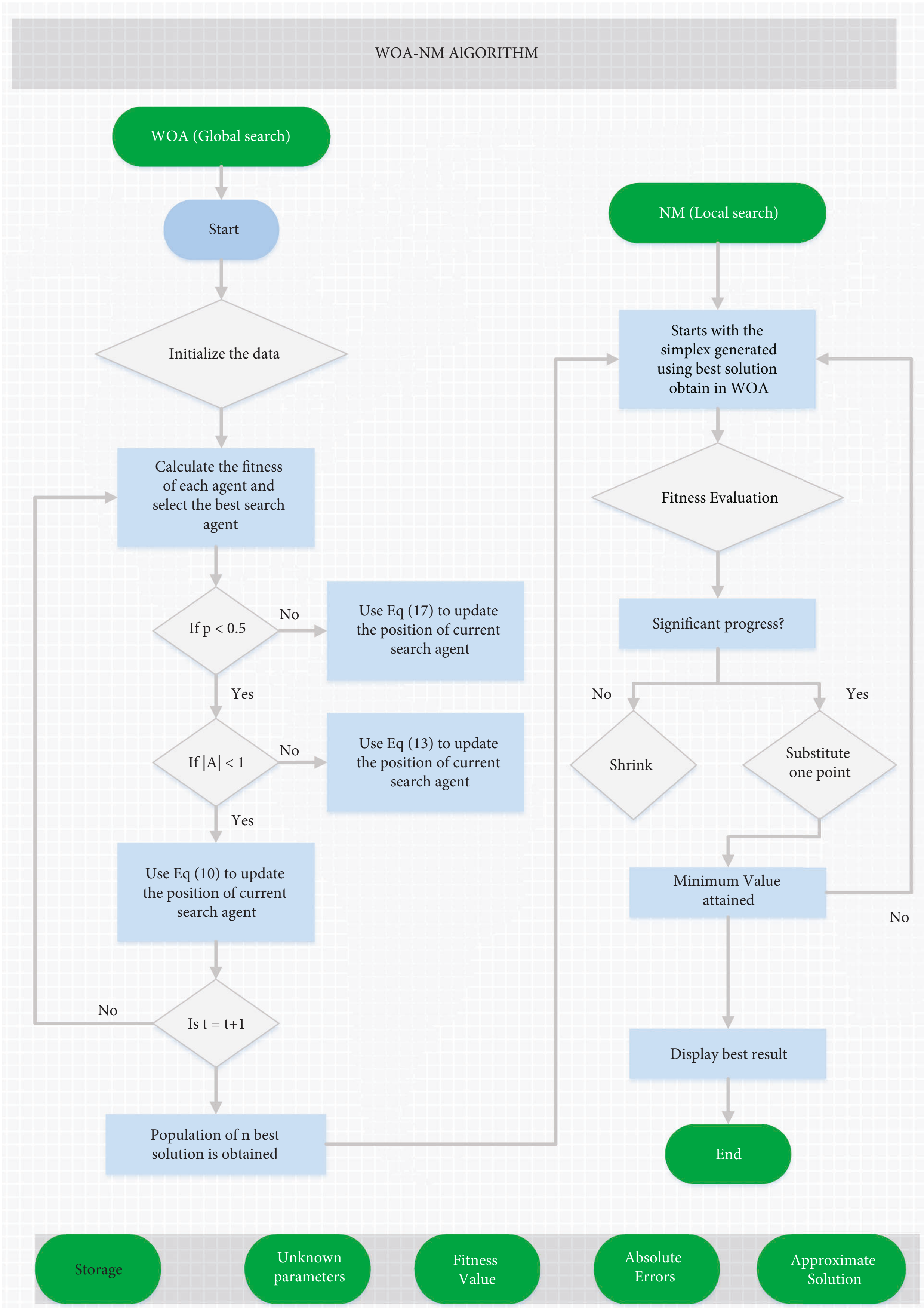


Figure 2: Graphical illustration of the working steps of the whale optimization algorithm and Nelder–Mead algorithm for the training of neurons in the LeNN architecture and the minimization of fitness functions.

$$[NSE_S, NSE_X, NSE_Y] = \begin{bmatrix} 1 - \dfrac{\sum_{i=1}^{n}\left(S(t_i) - S_{\text{approx}}(t_i)\right)^2}{\sum_{i=1}^{n}\left(S(t_i) - S\prime(t_i)\right)^2} & S\prime(t_i) = \dfrac{1}{n}\sum_{i=1}^{n} S(t_i) \\ 1 - \dfrac{\sum_{i=1}^{n}\left(X(t_i) - X_{\text{approx}}(t_i)\right)^2}{\sum_{i=1}^{n}\left(X(t_i) - X\prime(t_i)\right)^2} & X\prime(t_i) = \dfrac{1}{n}\sum_{i=1}^{n} X(t_i) \\ 1 - \dfrac{\sum_{i=1}^{n}\left(Y(t_i) - Y_{\text{approx}}(t_i)\right)^2}{\sum_{i=1}^{n}\left(Y(t_i) - Y\prime(t_i)\right)^2} & Y\prime(t_i) = \dfrac{1}{n}\sum_{i=1}^{n} Y(t_i) \end{bmatrix}^T, \tag{21}$$

$$[ENSE_S, ENSE_X, ENSE_Y] = [\,1 - NSE_S, 1 - NSE_X, 1 - NSE_Y\,], \tag{22}$$

where n shows total input grid points. The values of performance measures of MAD, RMSE, and ENSE should be equal to zero for perfect modeling, while NSE should be equal to 1. The global versions of MAD, RMSE, and ENSE for the given mathematical model of the prey-predator system are formulated by given equations:

$$[GMAD_S, GMAD_X, GMAD_Y] = \begin{bmatrix} \dfrac{1}{R}\sum_{r=1}^{R}\left(\dfrac{1}{n}\sum_{i=1}^{n}\left|S(t_i) - S_{\text{approx}}(t_i)\right|\right) \\ \dfrac{1}{R}\sum_{r=1}^{R}\left(\dfrac{1}{n}\sum_{i=1}^{n}\left|X(t_i) - X_{\text{approx}}(t_i)\right|\right) \\ \dfrac{1}{R}\sum_{r=1}^{R}\left(\dfrac{1}{n}\sum_{i=1}^{n}\left|Y(t_i) - Y_{\text{approx}}(t_i)\right|\right) \end{bmatrix}^T, \tag{23}$$

$$[GTIC_S, GTIC_X, GTIC_Y] = \begin{bmatrix} \dfrac{1}{R}\sum_{r=1}^{R}\left(\dfrac{\sqrt{(1/n)\sum_{i=1}^{n}\left(S(t_i) - S_{\text{approx}}(t_i)\right)^2}}{\sqrt{(1/n)\sum_{i=1}^{n}\left(S(t_i)\right)^2} + \sqrt{(1/n)\sum_{i=1}^{n}\left(S_{\text{approx}}(t_i)\right)^2}}\right) \\ \dfrac{1}{R}\sum_{r=1}^{R}\left(\dfrac{\sqrt{(1/n)\sum_{i=1}^{n}\left(X(t_i) - X_{\text{approx}}(t_i)\right)^2}}{\sqrt{(1/n)\sum_{i=1}^{n}\left(X(t_i)\right)^2} + \sqrt{(1/n)\sum_{i=1}^{n}\left(X_{\text{approx}}(t_i)\right)^2}}\right) \\ \dfrac{1}{R}\sum_{r=1}^{R}\left(\dfrac{\sqrt{(1/n)\sum_{i=1}^{n}\left(Y(t_i) - Y_{\text{approx}}(t_i)\right)^2}}{\sqrt{(1/n)\sum_{i=1}^{n}\left(Y(t_i)\right)^2} + \sqrt{(1/n)\sum_{i=1}^{n}\left(Y_{\text{approx}}(t_i)\right)^2}}\right) \end{bmatrix}^T, \tag{24}$$

$$[GNSE_S, GNSE_X, GNSE_Y] = \begin{bmatrix} \frac{1}{R}\sum_{r=1}^{R}\left(1-\frac{\sum_{i=1}^{n}\left(S(t_i)-S_{\text{approx}}(t_i)\right)^2}{\sum_{i=1}^{n}\left(S(t_i)-S\prime(t_i)\right)^2}\right) & S\prime(t_i)=\frac{1}{n}\sum_{i=1}^{n}S(t_i) \\ \frac{1}{R}\sum_{r=1}^{R}\left(1-\frac{\sum_{i=1}^{n}\left(X(t_i)-X_{\text{approx}}(t_i)\right)^2}{\sum_{i=1}^{n}\left(X(t_i)-X\prime(t_i)\right)^2}\right) & X\prime(t_i)=\frac{1}{n}\sum_{i=1}^{n}X(t_i) \\ \frac{1}{R}\sum_{r=1}^{R}\left(1-\frac{\sum_{i=1}^{n}\left(Y(t_i)-Y_{\text{approx}}(t_i)\right)^2}{\sum_{i=1}^{n}\left(Y(t_i)-Y\prime(t_i)\right)^2}\right) & Y\prime(t_i)=\frac{1}{n}\sum_{i=1}^{n}Y(t_i) \end{bmatrix}^T, \tag{25}$$

$$[GENSE_S, GENSE_X, GENSE_Y] = [\, 1-GNSE_S, 1-GNSE_X, 1-GNSE_Y \,], \tag{26}$$

where R denotes the number of independent runs. Global fitness (GFIT) is defined as the mean of fitness values attained in independent runs. Mathematically, GFIT is given as follows:

$$GFIT = \frac{1}{R}\sum_{r=1}^{R}\epsilon_r. \tag{27}$$

7. Result and Discussion

The numerical solutions of the prey-predator system with immigrant prey under the influence of variations in various parameters are investigated with the proposed methodology. The exact solution for these nonlinear models is not known, so the comparative study is conducted with MATLAB solver ode45 and the homotopy perturbation method [44]. Six problems of the prey-predator model are considered for different cases depending on values of coefficients, i.e, $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1$, and γ_2. A graphical abstract of the paper is shown in Figure 3.

7.1. Problem I: Influence of Variations in α_1 on the Prey-Predator Model. In this problem, the effect of variations in the intrinsic growth rate of local prey α_1 on population density is discussed. An error-based fitness function along with initial populations is given as follows:

$$\text{Minimize}\,\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5 + \epsilon_6, \tag{28}$$

where ϵ_1 to ϵ_6 are defined by

$$\begin{cases} \epsilon_1 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{34}\left(\frac{dS_n}{dt} - S_n\left(\alpha_1 - \frac{\alpha_1 S_n}{50}\right) - (0.2)S_nX_n + (0.01)S_nY_n\right)^2, \\ \epsilon_2 = \dfrac{1}{N}\displaystyle\sum_{n=35}^{68}\left(\frac{dX_n}{dt} - X_n\left((0.2) - \frac{(0.2)X_n}{k_2}\right) + (0.1)S_nX_n + (0.9)X_nY_n\right)^2, \\ \epsilon_3 = \dfrac{1}{N}\displaystyle\sum_{n=69}^{102}\left(\frac{dY_n}{dt} - (0.9)S_nY_n - (0.8)X_nY_n + (0.01)Y_n + (0.01)Y_n^2\right)^2, \\ \epsilon_4 = (S_0 - 0.5)^2, \\ \epsilon_5 = (X_0 - 0.5)^2, \\ \epsilon_6 = (Y_0 - 0.5)^2. \end{cases} \tag{29}$$

Five cases are considered, depending on the value of α_1.

Case I: $\alpha_1 = 0.06$

Case II: $\alpha_1 = 0.08$

Case III: $\alpha_1 = 0.10$

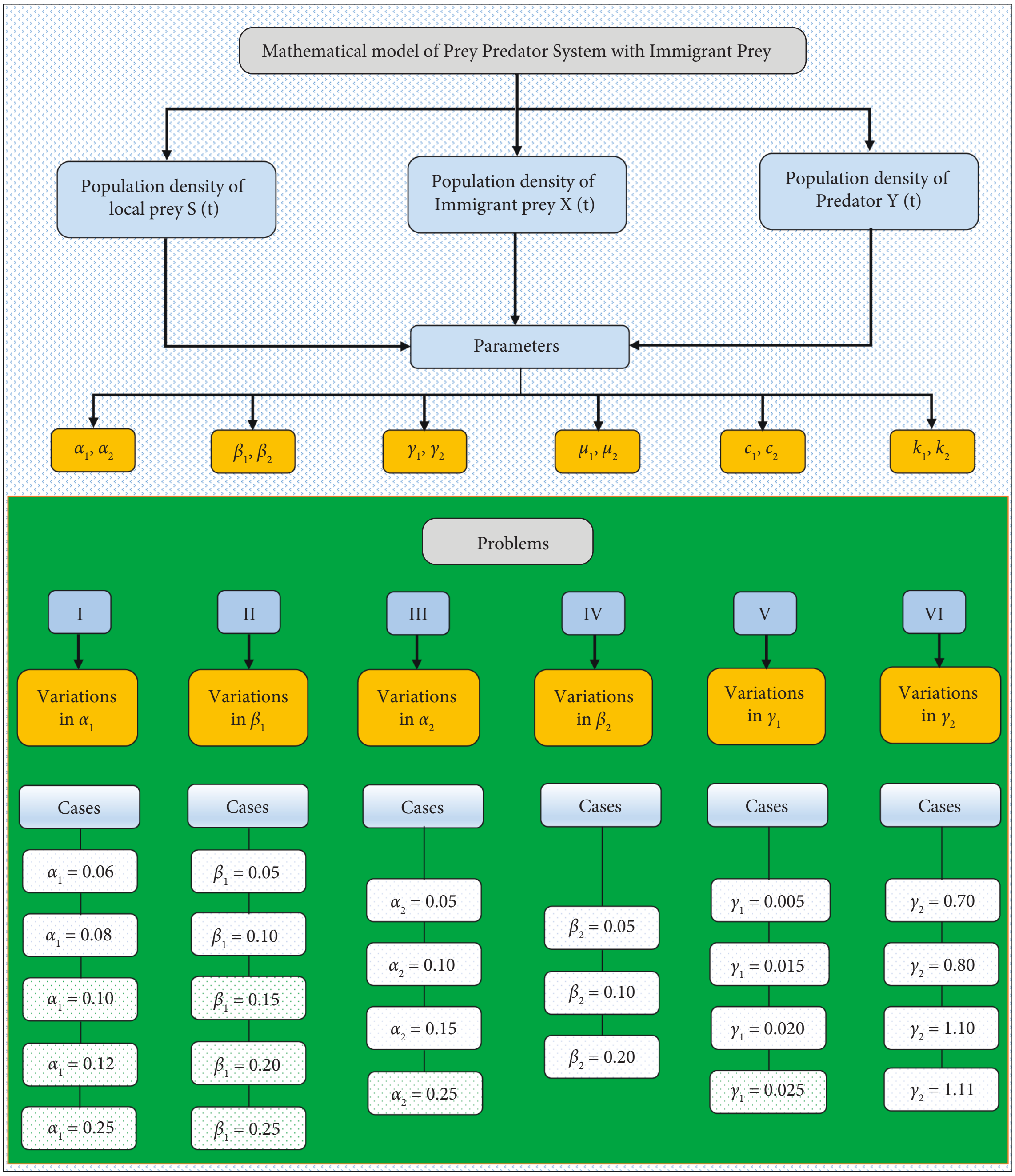


FIGURE 3: Graphical overview of the paper.

TABLE 2: Comparison between approximate solutions obtained by the LeNN-WOA-NM algorithm for the prey-predator system with the homotopy perturbation method [44] for $\alpha_1 = 0.12$.

t	$S(t)$		$X(t)$		$Y(t)$	
	HPO	LeNN-WOA-NM	HPO	LeNN-WOA-NM	HPO	LeNN-WOA-NM
0.0	0.500000	0.50000000	0.500000	0.50000000	0.500000	0.50000000
0.2	0.521458	0.52145956	0.466762	0.46676321	0.590440	0.59114765
0.4	0.542942	0.54294366	0.427969	0.42797117	0.695777	0.69922187
0.6	0.564244	0.56424422	0.384329	0.38433104	0.817468	0.82687284
0.8	0.585155	0.58515597	0.337008	0.33701139	0.956844	0.97704917
1.0	0.605478	0.60547960	0.287606	0.28761008	1.115127	1.15310441

Table 3: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population density of local prey $S(t)$ under the influence of the intrinsic growth rate of local prey α_1.

t	$\alpha_1 = 0.06$	$\alpha_1 = 0.08$	$\alpha_1 = 0.10$	$\alpha_1 = 0.12$	$\alpha_1 = 0.25$
0.0	0.50000029	0.50000024	0.50000001	0.50000000	0.50000000
0.2	0.51530461	0.51734394	0.51940166	0.52145956	0.53504743
0.4	0.53020804	0.53441582	0.53866566	0.54294366	0.57158810
0.6	0.54451936	0.55101532	0.55759565	0.56424422	0.60943884
0.8	0.55806566	0.56695084	0.57598624	0.58515597	0.64838425
1.0	0.57069056	0.58205878	0.59365221	0.60547960	0.68819372

Table 4: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population density of immigrant prey $X(t)$ under the influence of the intrinsic growth rate of local prey α_1.

t	$\alpha_1 = 0.06$	$\alpha_1 = 0.08$	$\alpha_1 = 0.10$	$\alpha_1 = 0.12$	$\alpha_1 = 0.25$
0.0	0.49999979	0.49999997	0.49999999	0.50000000	0.50000000
0.2	0.46679789	0.46677660	0.46677363	0.46676321	0.46668320
0.4	0.42814941	0.42808998	0.42803011	0.42797117	0.42757426
0.6	0.38480296	0.38463971	0.38448897	0.38433104	0.38327075
0.8	0.33796050	0.33764489	0.33732977	0.33701139	0.33485636
1.0	0.28923060	0.28868911	0.28815986	0.28761008	0.28389875

Table 5: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population density of the predator $Y(t)$ under the influence of the intrinsic growth rate of local prey α_1.

t	$\alpha_1 = 0.06$	$\alpha_1 = 0.08$	$\alpha_1 = 0.10$	$\alpha_1 = 0.12$	$\alpha_1 = 0.25$
0.0	0.49999991	0.49999919	0.49999999	0.50000000	0.50000000
0.2	0.59011694	0.59022292	0.59032756	0.59047007	0.59114765
0.4	0.69423291	0.69474410	0.69525628	0.69577735	0.69922187
0.6	0.81332517	0.81468984	0.81607383	0.81746843	0.82687284
0.8	0.94811217	0.95097827	0.95389198	0.95684491	0.97704917
1.0	1.09907344	1.10432082	1.10967602	1.11512837	1.15310441

Table 6: Absolute errors obtained by the LeNN-WOA-NM algorithm for population density of local prey $S(t)$ under the influence of the intrinsic growth rate of local prey α_1.

t	$\alpha_1 = 0.06$	$\alpha_1 = 0.08$	$\alpha_1 = 0.10$	$\alpha_1 = 0.12$	$\alpha_1 = 0.25$
0.0	$4.25E-10$	$9.71E-11$	$4.27E-11$	$9.01E-11$	$3.05E-13$
0.2	$3.87E-10$	$3.87E-11$	$1.80E-09$	$8.37E-11$	$9.38E-14$
0.4	$5.99E-11$	$1.16E-10$	$1.85E-09$	$3.57E-11$	$3.30E-13$
0.6	$1.46E-10$	$1.33E-10$	$2.21E-09$	$3.05E-11$	$1.05E-13$
0.8	$1.37E-10$	$2.15E-11$	$3.80E-09$	$7.04E-11$	$1.82E-12$
1.0	$9.93E-11$	$2.11E-10$	$6.82E-11$	$6.49E-11$	$4.90E-14$

Case IV: $\alpha_1 = 0.12$

Case V: $\alpha_1 = 0.25$

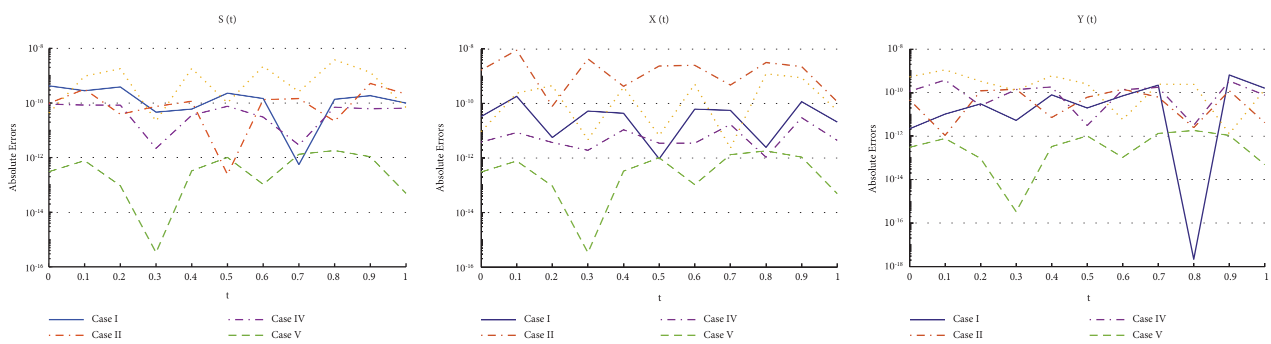
The LeNN-WOA-NM algorithm is applied to prey-predator model equation (29) to study the influence of variations in the intrinsic growth rate of local prey. A comparison between approximate solutions obtained by the LeNN-WOA-NM algorithm for the prey-predator system with the homotopy perturbation method [44] for $\alpha_1 = 0.12$ is illustrated in Table 2. Approximate solutions for population densities of local prey, immigrant prey, and predator are given in Tables 3–5, respectively. Absolute errors are

Table 7: Absolute errors obtained by the LeNN-WOA-NM algorithm for population density of immigrant prey $X(t)$ under the influence of the intrinsic growth rate of local prey α_1.

t	$\alpha_1 = 0.06$	$\alpha_1 = 0.08$	$\alpha_1 = 0.10$	$\alpha_1 = 0.12$	$\alpha_1 = 0.25$
0.0	$3.33E-11$	$1.68E-09$	$9.24E-12$	$3.85E-12$	$3.05E-13$
0.2	$5.66E-12$	$7.98E-11$	$4.55E-10$	$3.72E-12$	$9.38E-14$
0.4	$4.40E-11$	$4.29E-10$	$4.03E-10$	$1.09E-11$	$3.30E-13$
0.6	$6.22E-11$	$2.53E-09$	$5.15E-10$	$3.52E-12$	$1.05E-13$
0.8	$2.46E-12$	$3.20E-09$	$1.24E-09$	$1.07E-12$	$1.82E-12$
1.0	$2.09E-11$	$1.17E-10$	$5.55E-11$	$4.44E-12$	$4.90E-14$

Table 8: Absolute errors obtained by the LeNN-WOA-NM algorithm for population density of the predator $Y(t)$ under the influence of the intrinsic growth rate of local prey α_1.

t	$\alpha_1 = 0.06$	$\alpha_1 = 0.08$	$\alpha_1 = 0.10$	$\alpha_1 = 0.12$	$\alpha_1 = 0.25$
0.0	$2.17E-12$	$4.68E-11$	$5.32E-10$	$1.16E-10$	$3.05E-13$
0.2	$3.04E-11$	$1.21E-10$	$3.32E-10$	$2.52E-11$	$9.38E-14$
0.4	$7.91E-11$	$7.16E-12$	$5.91E-10$	$1.83E-10$	$3.30E-13$
0.6	$7.12E-11$	$1.43E-10$	$5.80E-12$	$1.36E-10$	$1.05E-13$
0.8	$2.17E-18$	$2.46E-12$	$2.41E-10$	$3.16E-12$	$1.82E-12$
1.0	$1.60E-10$	$4.16E-12$	$1.40E-10$	$7.66E-11$	$4.90E-14$



(a) (b) (c)

Figure 4: Absolute errors obtained by the LeNN-WOA-NM algorithm under the influence of variations in the intrinsic growth rate of local prey on population densities of the prey-predator model. (a) Population density of local prey. (b) Population density of immigrant prey. (c) Population density of the predator.

presented in Tables 6–8 and are graphically illustrated in Figure4. Table 9 represents the statistics of global values of performance indicators during 100 independent trails. Figure 5 shows the convergence of the fitness value. It can be seen that the fitness value for each case lies around 10^{-6} to 10^{-10}. The percentage convergence of the fitness value and performance indicators during multiple runs is shown in Table 10. Trained neurons in the LeNN structure for obtaining best solutions are shown in Table 11. From Figure 6, the following conclusions are drawn:

(i) Population density of local prey has a direct relation with the intrinsic growth rate of local prey

(ii) Population density of immigrant prey has an inverse relation with the intrinsic growth rate of local prey

(iii) Population density of the predator varies directly with the intrinsic growth rate of local prey

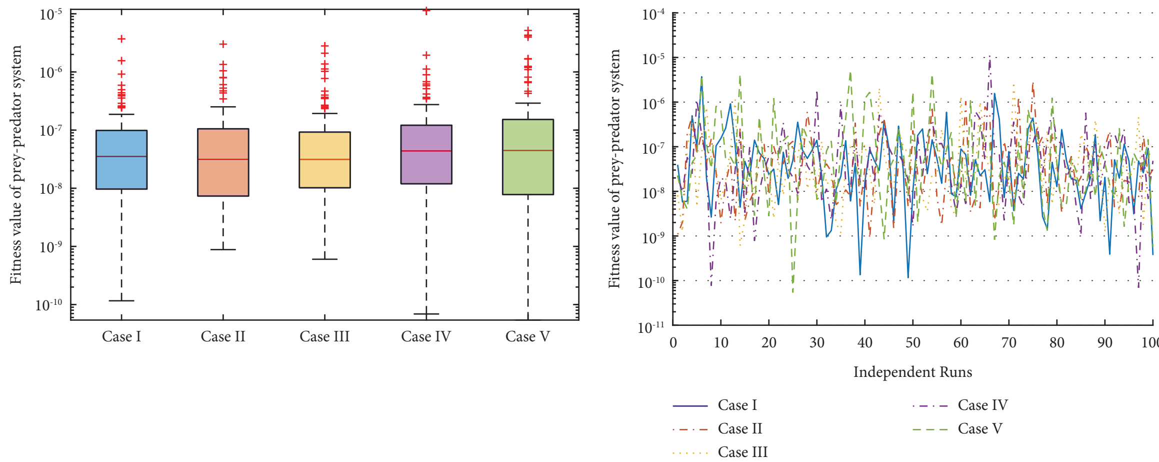
7.2. Problem II: Effect of Variations in β_1 on the Prey-Predator Model. In this problem, the effect of variations in the positive impact of force of interaction β_1 between local and immigrant prey on population densities of the prey-predator model is discussed. An error-based fitness function along with initial populations is given as follows:

$$\text{Minimize}\, \epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5 + \epsilon_6, \tag{30}$$

where ϵ_1 to ϵ_6 are defined as follows:

Table 9: Statistics of global performance indices for variations in the intrinsic growth rate of local prey on the prey-predator model.

Cases	$S(t)$				$X(t)$				$Y(t)$			
	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE
I	$1.21E-07$	$2.47E-05$	$1.30E-05$	$3.82E-05$	$1.21E-07$	$2.26E-05$	$1.61E-05$	$3.21E-06$	$1.55E-07$	$3.70E-05$	$1.30E-05$	$2.00E-06$
II	$5.15E-08$	$1.64E-05$	$8.52E-06$	$8.13E-06$	$5.25E-08$	$1.69E-05$	$1.19E-05$	$1.15E-06$	$5.66E-08$	$2.67E-05$	$9.42E-06$	$3.87E-07$
III	$8.12E-07$	$2.74E-05$	$1.39E-05$	$2.61E-05$	$1.45E-07$	$2.14E-05$	$1.51E-05$	$2.90E-06$	$1.61E-07$	$3.89E-05$	$1.40E-05$	$1.15E-06$
IV	$5.57E-07$	$1.88E-05$	$9.59E-06$	$7.77E-06$	$5.86E-08$	$1.62E-05$	$1.17E-05$	$1.19E-06$	$9.01E-08$	$3.19E-05$	$1.14E-05$	$5.85E-07$
V	$6.95E-07$	$2.20E-05$	$1.02E-05$	$3.59E-06$	$1.05E-07$	$1.99E-05$	$1.43E-05$	$2.04E-06$	$1.37E-07$	$3.72E-05$	$1.30E-05$	$7.78E-07$



(a) (b)

Figure 5: Comparison between the box plots and the convergence graph of fitness evaluation over 100 independent runs for the prey-predator model with variations in the intrinsic growth rate of local prey. (a) Box plots for fitness evaluation of the prey-predator model with variations in α_1 (b) Convergence of the fitness value of the prey-predator model with variations in α_1.

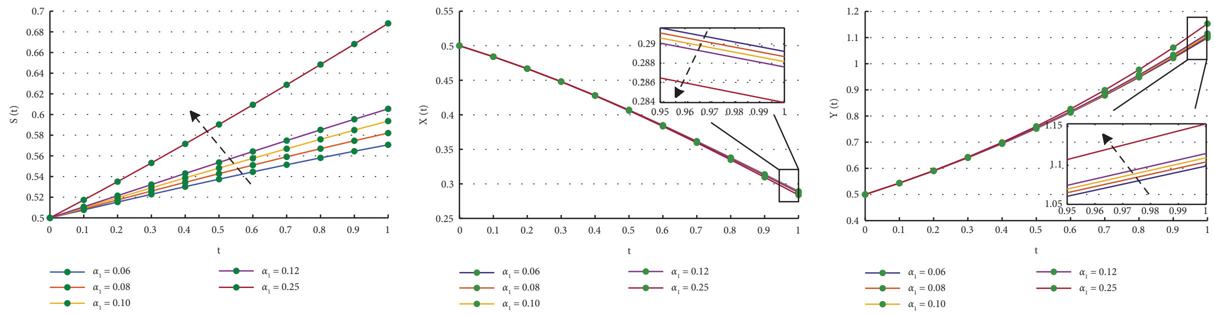
Table 10: Convergence analysis for variation in α_1 on the prey-predator model.

Cases		FIT				MAD			TIC			ENSE		
		$\leq 10^{-7}$	$\leq 10^{-8}$	$\leq 10^{-9}$	$\leq 10^{-10}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-6}$	$\leq 10^{-7}$	$\leq 10^{-8}$
$S(t)$	I	98	75	25	5	100	99	35	100	99	61	39	22	3
	II	80	30	3	0	100	100	49	100	100	56	68	36	8
	III	78	34	2	0	100	94	41	100	99	63	74	33	3
	IV	81	37	4	0	100	99	43	100	100	68	83	44	16
	V	77	27	5	0	100	97	42	100	100	66	87	60	27
$X(t)$	I	81	37	13	3	96	44	2	98	54	3	65	23	6
	II	85	41	11	0	100	43	1	100	100	58	71	20	3
	III	74	41	17	1	97	39	0	100	98	52	64	24	8
	IV	86	47	16	4	100	46	4	100	100	55	73	37	9
	V	80	47	12	1	98	43	3	100	99	57	68	26	6
$Y(t)$	I	80	41	16	3	100	95	21	99	57	9	47	14	7
	II	86	47	9	1	100	98	28	100	68	5	46	13	0
	III	78	36	8	2	100	91	20	99	57	4	45	10	0
	IV	76	42	9	3	100	94	26	100	60	8	44	10	6
	V	70	29	8	1	100	95	21	100	57	7	41	16	1

$$\begin{cases} \epsilon_1 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{34}\left(\dfrac{\mathrm{d}S_n}{\mathrm{d}t} - S_n\left((0.12) - \dfrac{(0.12)S_n}{50}\right) - (\beta_1)S_nX_n + (0.01)S_nY_n\right)^2, \\ \epsilon_2 = \dfrac{1}{N}\displaystyle\sum_{n=35}^{68}\left(\dfrac{\mathrm{d}X_n}{\mathrm{d}t} - X_n\left((0.2) - \dfrac{(0.2)X_n}{k_2}\right) + (0.1)S_nX_n + (0.9)X_nY_n\right)^2, \\ \epsilon_3 = \dfrac{1}{N}\displaystyle\sum_{n=69}^{102}\left(\dfrac{\mathrm{d}Y_n}{\mathrm{d}t} - (0.9)S_nY_n - (0.8)X_nY_n + (0.01)Y_n + (0.01)Y_n^2\right)^2, \\ \epsilon_4 = (S_0 - 0.5)^2, \\ \epsilon_5 = (X_0 - 0.5)^2, \\ \epsilon_6 = (Y_0 - 0.5)^2. \end{cases} \tag{31}$$

Table 11: Unknown parameters obtained by the LeNN-WOA-NM algorithm for the prey-predator model under the influence of variations in the intrinsic growth rate of local prey α_1.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
Case I	1	0.24298626	−0.2668655	0.09328582	0.17524653	0.0657111	0.45518402	0.62050563	0.107119458	−0.184535
	2	0.12478022	0.39026984	0.13404058	−0.1205176	−0.504835	−0.152368	0.29193615	−0.06286814	0.043174485
	3	0.15946384	0.47205052	0.04884412	−0.0142591	−0.0346883	0.98886623	0.09715236	0.181551811	0.868732734
	4	0.01120683	0.24610287	−0.2837988	0.14946792	0.16097393	−0.074499	0.46391737	−0.16073035	0.113397979
	5	−0.022669	0.0943905	0.0992974	0.34191594	0.58292848	0.00959628	0.06797262	−0.11244358	0.013444379
	6	−0.3216158	0.1410155	0.18325041	0.03228638	0.70287879	0.12845754	−0.1361841	0.158063474	−0.05836861
	7	0.48768209	0.0293772	0.17094612	−0.0018354	0.14860632	0.65130564	0.67663964	0.314984108	0.261128949
	8	0.08329406	0.04117897	−0.0205859	−0.1064367	−0.1326281	−0.0481534	−0.0486244	−0.33359642	0.564727586
	9	0.87211207	−0.0743821	0.18882808	0.69221376	−0.1213543	0.01584571	0.00730463	−0.07959673	0.248981123
	10	0.74644658	0.01575884	0.17044653	0.13151592	0.13622307	0.26101806	−0.1087744	−0.09419355	−32.2098458
	11	0.26313561	0.09923455	−0.0506162	0.01009165	0.13795634	0.05968636	0.42506235	0.302798399	0.094716386
Case II	1	0.33259121	−0.3657449	0.21062276	0.25900442	0.15918437	0.41026629	0.29040796	−1.70842597	−0.22039024
	2	0.17889024	−0.0431343	−0.6411116	0.23110108	0.42625725	0.08496042	−0.0683335	−0.01537247	0.252533232
	3	0.07297637	0.1667513	−0.0811545	−0.2048892	−0.1120065	−0.0826385	−0.6647359	−0.16394699	−0.00997236
	4	−0.1365861	0.47090897	−0.0639777	$3.43E-05$	0.1546024	−0.178235	0.4646494	0.190521738	0.195440966
	5	−0.007358	0.13102291	0.24249896	−0.1088503	−0.1671249	0.16954851	−0.7315874	−0.03059029	−0.07106493
	6	0.43362447	−0.155685	0.44563542	0.22079963	−0.6459135	−0.0878247	−0.5471167	−0.36068591	−0.90966495
	7	−0.4072594	0.43108895	−0.0973015	0.00859421	0.19519171	0.05831302	0.00294865	−0.03440871	0.372338435
	8	−0.015323	−0.0127997	−0.079715	0.41948941	−0.2878455	0.21735868	−0.839503	−0.04815065	0.025011228
	9	0.26217328	0.20028492	0.2222801	−0.1630391	0.17520391	0.0974704	0.34891726	−0.13212935	−0.1419514
	10	0.09242525	0.19371262	−0.1574267	0.34189851	0.28434772	0.08494461	0.46694404	−0.12329874	39.39674102
	11	0.13225195	−0.0846055	−0.0281568	0.14217372	0.31887231	0.42910377	0.26716207	0.088173159	0.085561799
Case III	1	0.08792719	−0.02998	1.2174462	0.69165686	0.06374932	−0.0488322	0.11491028	0.090075909	−0.32604325
	2	0.11610049	1.56406569	0.0795106	−0.0902964	0.09097774	−0.1017928	0.06371619	0.492255996	0.406557357
	3	0.03262421	−0.2983161	0.4361223	0.71023061	0.27130756	0.53574029	0.74546663	0.731579954	−0.11925334
	4	−0.8534256	0.27102507	0.09970798	−0.4630868	0.19588223	0.25655668	−0.1733402	0.4924538	−0.02738845
	5	−0.4373573	0.06892556	0.11113751	0.22424973	0.18678342	−0.0095592	0.13860986	0.342147207	−0.12816846
	6	0.85585046	−0.0086572	0.70804298	0.16532453	0.46387591	0.19167055	−0.4155845	0.358174513	0.519586436
	7	0.4731327	−0.0131084	0.53818072	0.00107541	−0.2371714	0.21538868	0.27067076	0.007722887	−0.11651201
	8	−0.1730569	−0.0542656	1.00463015	0.33304716	0.42111507	0.54031591	−0.0982948	−0.30964889	0.225264957
	9	0.01950551	0.3195719	−0.1781725	0.00222254	−0.0527906	0.33879162	0.14536108	0.185060402	0.022547703
	10	−0.2758316	0.27171296	−0.0428651	0.07565986	0.29894632	−0.1182305	0.19684946	0.626442716	0.062105445
	11	0.28574702	−0.2628512	0.06102257	0.04403502	0.22414615	0.40364307	0.55206157	0.429077344	−0.00256072
Case IV	1	0.25369644	−0.4777306	−0.8361847	−0.0621264	0.27351167	−0.6065099	0.68147388	−1.43275825	−0.10932386
	2	0.06991497	−0.1139906	−0.4735192	−0.5854725	−0.1661107	0.07770566	−0.2460336	0.052462785	0.191077917
	3	−0.2351744	−0.4870632	−0.0716903	−0.2556044	$9.83E-05$	0.02156199	−0.3140772	−0.4208503	−0.13182798
	4	−0.0489805	0.03121692	−0.2705836	−0.2176759	−0.3388043	−0.1570093	−0.2947292	0.000861238	0.091135849
	5	0.12066069	0.00528409	−0.1876653	0.18129324	−0.2476741	−0.0843931	0.33223353	0.037877215	0.552486838
	6	−0.2917694	0.09885985	−0.655587	−0.5821936	0.02484774	−0.2678441	−0.1209825	0.019527337	−0.2041232
	7	0.2180106	0.10156035	−0.0839076	−0.0408137	0.20227179	0.08783601	−1.1472874	0.088467804	−0.1006823
	8	−0.3594382	−0.0329465	−0.0085748	−0.076245	0.071221	−0.2677479	−0.6912671	0.072673015	−0.60129414
	9	0.12647778	−0.0321146	−0.1925977	−0.1982251	−0.1306954	−0.4290151	−0.4035842	−0.01597822	−0.09388657
	10	−0.7218529	−0.127281	−0.253891	−0.520109	−0.148345	−0.1301849	0.45904036	−0.14555693	0.210878261
	11	−0.0415557	−0.0626802	−0.0135924	−0.2376142	−0.2602334	0.31718881	−0.0300309	−0.07780308	0.119273654
Case V	1	0.11535169	0.09331411	0.06459598	0.13708955	0.02523131	0.14440187	0.53678594	0.289671331	0.265496244
	2	1.2018728	0.66566882	−0.0195754	0.55314202	0.20920125	0.53535701	0.3092534	0.73571248	0.320987524
	3	0.20319881	−0.8256885	1.09540267	0.09636349	0.12623634	−0.1361461	0.29987708	−0.14730456	−0.1852485
	4	0.0272977	0.50122086	−0.2264851	−0.0410539	0.34541103	−0.1572514	0.88612589	0.219231183	0.296041826
	5	0.13436663	0.09124816	0.31502947	0.03670816	−0.2842128	0.28047538	0.30493881	0.228728298	−0.83846218
	6	0.59090535	0.2219209	0.35836933	0.13287532	0.07279467	0.5746159	0.05242013	0.22070811	0.059736502
	7	0.57147768	0.17459027	0.74963519	0.96594705	0.18063006	−0.2505621	0.11584087	−0.02191061	0.160702925
	8	0.14083549	0.5151422	0.06170226	0.18189487	0.19412865	0.22132998	0.9645296	−0.15578224	0.422131366
	9	−0.0987012	−0.3447324	0.30547697	0.22752865	0.56631904	0.08438966	0.0392108	−0.15270349	0.520225963
	10	0.05337954	0.38978704	0.32410767	0.04067255	0.40603952	0.21526779	−0.0360576	−0.00016971	−1.80718751
	11	0.16882868	0.42247401	0.06696436	−0.0534127	0.38690363	0.40861018	−0.1032548	0.039874898	0.21583529



(a) (b) (c)

FIGURE 6: Comparison between solutions obtained by the LeNN-WOA-NM algorithm under the influence of variations in the intrinsic growth rate of local prey on population densities of the prey-predator model. (a) Approximate solutions for population density of local prey. (b) Approximate solutions for population density of immigrant prey. (c) Approximate solutions for population density of the predator.

TABLE 12: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population density of local prey $S(t)$ under the influence of β_1.

t	$\beta_1 = 0.05$	$\beta_1 = 0.10$	$\beta_1 = 0.15$	$\beta_1 = 0.20$	$\beta_1 = 0.25$
0.0	0.49999811	0.50000565	0.49999994	0.50000001	0.49999994
0.2	0.51394192	0.51644576	0.51893794	0.52145440	0.52399164
0.4	0.52796160	0.53293741	0.53790411	0.54293644	0.54801843
0.6	0.54203480	0.54936883	0.55674985	0.56424287	0.57183842
0.8	0.55610994	0.56564641	0.57531194	0.58515293	0.59516204
1.0	0.57009452	0.58167704	0.59345033	0.60547330	0.61773135

TABLE 13: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population density of immigrant prey $X(t)$ under the influence of β_1.

t	$\beta_1 = 0.05$	$\beta_1 = 0.10$	$\beta_1 = 0.15$	$\beta_1 = 0.20$	$\beta_1 = 0.25$
0.0	0.49999992	0.50000514	0.49999988	0.50000001	0.49999969
0.2	0.46680042	0.46679364	0.46677626	0.46676260	0.46673440
0.4	0.42817965	0.42812547	0.42804193	0.42797008	0.42789521
0.6	0.38488898	0.38470693	0.38451903	0.38432922	0.38413261
0.8	0.33810661	0.33775597	0.33737496	0.33701285	0.33663106
1.0	0.28943501	0.28883454	0.28823065	0.28760701	0.28697657

TABLE 14: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population density of the predator $Y(t)$ under the influence of β_1.

t	$\beta_1 = 0.05$	$\beta_1 = 0.10$	$\beta_1 = 0.15$	$\beta_1 = 0.20$	$\beta_1 = 0.25$
0.0	0.50000032	0.49999215	0.50000005	0.50000003	0.49999973
0.2	0.59004112	0.59016271	0.59030647	0.59044066	0.59057741
0.4	0.69390817	0.69451560	0.69516439	0.69578033	0.69640936
0.6	0.81260140	0.81419598	0.81584261	0.81746743	0.81912365
0.8	0.94690117	0.95016524	0.95348538	0.95684239	0.96025853
1.0	1.09742754	1.10322425	1.10912457	1.11513013	1.12125476

Table 15: Absolute errors obtained by the LeNN-WOA-NM algorithm for population density of local prey $S(t)$ under the influence of β_1.

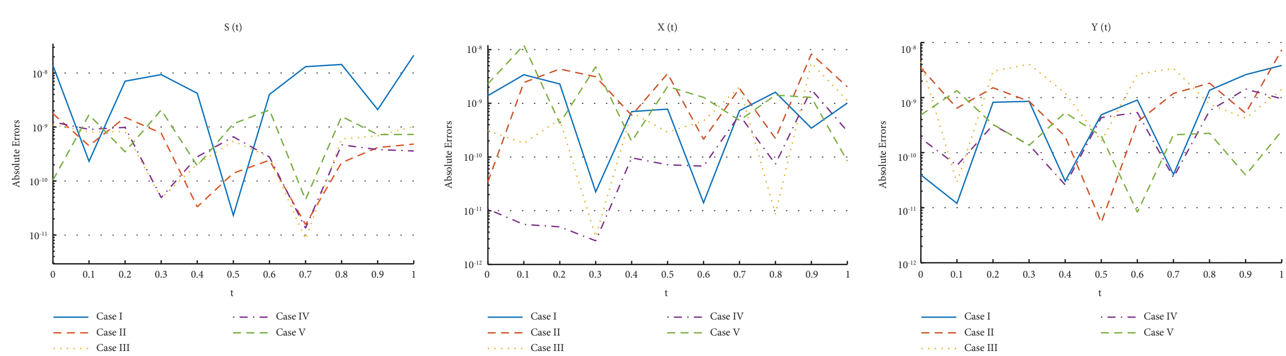
t	$\beta_1 = 0.05$	$\beta_1 = 0.10$	$\beta_1 = 0.15$	$\beta_1 = 0.20$	$\beta_1 = 0.25$
0.0	$1.38E-08$	$1.77E-09$	$1.10E-09$	$1.24E-09$	$1.04E-10$
0.2	$7.06E-09$	$1.51E-09$	$8.29E-10$	$9.85E-10$	$3.47E-10$
0.4	$4.21E-09$	$3.33E-11$	$2.05E-10$	$2.81E-10$	$1.92E-10$
0.6	$4.01E-09$	$2.43E-10$	$2.77E-10$	$2.79E-10$	$2.07E-09$
0.8	$1.44E-08$	$2.19E-10$	$6.01E-10$	$4.68E-10$	$1.57E-09$
1.0	$2.13E-08$	$4.83E-10$	$1.02E-09$	$3.60E-10$	$7.30E-10$

Table 16: Absolute errors obtained by the LeNN-WOA-NM algorithm for population density of immigrant prey $X(t)$ under the influence of β_1.

t	$\beta_1 = 0.05$	$\beta_1 = 0.10$	$\beta_1 = 0.15$	$\beta_1 = 0.20$	$\beta_1 = 0.25$
0.0	$1.39E-09$	$3.44E-11$	$3.07E-10$	$1.07E-11$	$2.30E-09$
0.2	$2.29E-09$	$4.35E-09$	$5.04E-10$	$5.02E-12$	$4.24E-10$
0.4	$7.01E-10$	$5.98E-10$	$6.77E-10$	$9.67E-11$	$1.96E-10$
0.6	$1.41E-11$	$2.14E-10$	$4.69E-10$	$6.77E-11$	$1.29E-09$
0.8	$1.61E-09$	$2.21E-10$	$8.72E-12$	$7.39E-11$	$1.42E-09$
1.0	$1.01E-09$	$2.03E-09$	$1.13E-09$	$3.02E-10$	$8.42E-11$

Table 17: Absolute errors obtained by the LeNN-WOA-NM algorithm for population density of the predator $Y(t)$ under the influence of β_1.

t	$\beta_1 = 0.05$	$\beta_1 = 0.10$	$\beta_1 = 0.15$	$\beta_1 = 0.20$	$\beta_1 = 0.25$
0.0	$3.95E-11$	$3.48E-09$	$4.48E-09$	$1.83E-10$	$4.83E-10$
0.2	$8.22E-10$	$1.51E-09$	$3.01E-09$	$3.19E-10$	$3.25E-10$
0.4	$3.03E-11$	$1.96E-10$	$1.20E-09$	$2.60E-11$	$5.34E-10$
0.6	$9.03E-10$	$3.60E-10$	$2.64E-09$	$5.37E-10$	$8.30E-12$
0.8	$1.36E-09$	$1.83E-09$	$7.57E-10$	$5.75E-10$	$2.25E-10$
1.0	$3.84E-09$	$7.37E-09$	$1.40E-09$	$9.52E-10$	$2.47E-10$



(a) (b) (c)

Figure 7: Absolute errors obtained by the LeNN-WOA-NM algorithm under the influence of variations in β_1 on population densities of the prey-predator model. (a) Population density of local prey. (b) Population density of immigrant prey. (c) Population density of the predator.

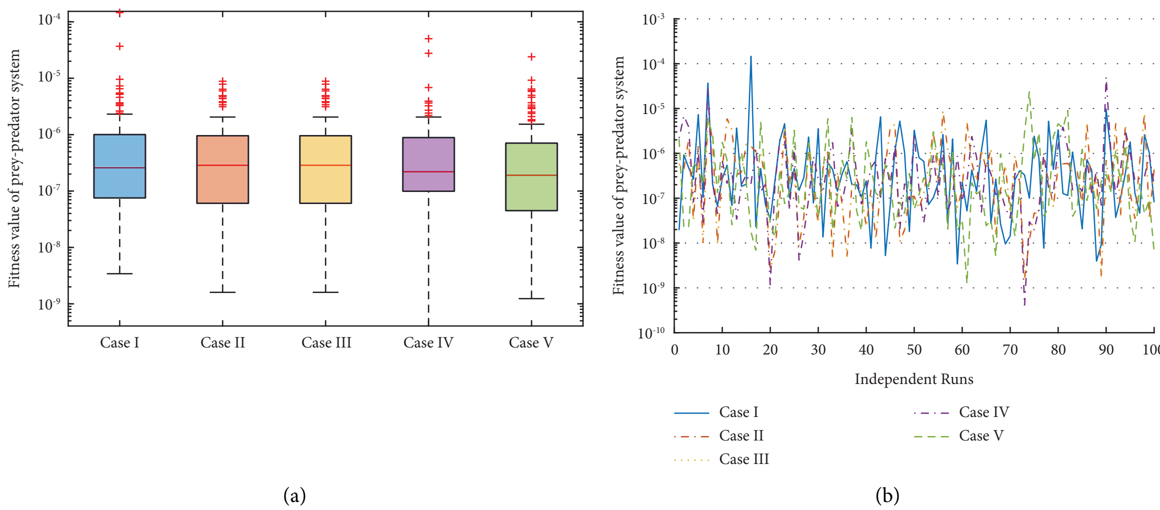


FIGURE 8: Comparison between the box plots and convergence graphs of fitness evaluation over 100 independent runs for the prey-predator model with variations in force of interaction between local and immigrant prey. (a) Box plots for fitness evaluation of the prey-predator model with variations in β_1. (b) Convergence of the fitness value of the prey-predator model with variations in β_1.

Five cases are considered, depending on the value of β_1.

Case I: $\beta_1 = 0.05$

Case II: $\beta_1 = 0.10$

Case III: $\beta_1 = 0.15$

Case IV: $\beta_1 = 0.20$

Case V: $\beta_1 = 0.25$

Approximate solutions for the effect of variations in β_1 on population densities of the prey-predator model are given in Tables 12–14. Absolute errors in our solution for population densities are presented in Tables 15–17, respectively. The solution of the design scheme overlaps the exact solution with absolute errors that lie between 10^{-9} and 10^{-11} as illustrated in Figure 7. Figure 8 shows the behavior of fitness evaluation for the prey-predator model under the influence of β_1. Convergence analysis of performance measures is given in Tables 18 and 19. Trained neurons obtained by the LeNN-WOA-NM algorithm for different cases of Eq (31) are shown in Table 20. From the solutions (see Figure 9), the following conclusions can be drawn:

(i) The negative force of interaction between local and immigrant prey has a direct relation with population density of local prey and the predator, while population density is inversely related to β_1

7.3. Problem III: Effect of Variations in α_2 on the Prey-Predator Model. In this problem, the effect of variations in the increasing rate of immigrant prey α_2 on population densities of the prey-predator model is discussed. An error-based fitness function along with initial populations is given as follows:

$$\text{Minimize}\ \epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5 + \epsilon_6, \tag{32}$$

where ϵ_1 to ϵ_6 are defined as follows:

$$\begin{cases} \epsilon_1 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{34}\left(\dfrac{dS_n}{dt} - S_n\left((0.12) - \dfrac{(0.12)S_n}{50}\right) - (0.2)S_nX_n + (0.01)S_nY_n\right)^2, \\ \epsilon_2 = \dfrac{1}{N}\displaystyle\sum_{n=35}^{68}\left(\dfrac{dX_n}{dt} - X_n\left(\alpha_2 - \dfrac{\alpha_2 X_n}{k_2}\right) + (0.1)S_nX_n + (0.9)X_nY_n\right)^2, \\ \epsilon_3 = \dfrac{1}{N}\displaystyle\sum_{n=69}^{102}\left(\dfrac{dY_n}{dt} - (0.9)S_nY_n - (0.8)X_nY_n + (0.01)Y_n + (0.01)Y_n^2\right)^2, \\ \epsilon_4 = (S_0 - 0.5)^2, \\ \epsilon_5 = (X_0 - 0.5)^2, \\ \epsilon_6 = (Y_0 - 0.5)^2. \end{cases} \tag{33}$$

Table 18: Comparison through global performance indices for variations in β_1 on the prey-predator model.

Cases	$S(t)$				$X(t)$				$Y(t)$			
	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE
I	8.39$E-06$	7.18$E-05$	3.75$E-05$	4.26$E-04$	2.61$E-06$	2.74$E-04$	2.14$E-04$	2.14$E-04$	1.63$E-06$	1.21$E-04$	4.34$E-05$	1.12$E-05$
II	1.00$E-06$	6.17$E-05$	3.16$E-05$	1.30$E-04$	8.45$E-07$	6.17$E-05$	4.37$E-05$	1.99$E-05$	1.54$E-06$	1.23$E-04$	4.33$E-05$	1.09$E-05$
III	1.64$E-06$	7.26$E-05$	3.67$E-05$	1.93$E-04$	2.27$E-06$	7.43$E-05$	5.27$E-05$	4.08$E-05$	1.49$E-06$	1.41$E-04$	4.91$E-05$	2.10$E-05$
IV	1.44$E-06$	6.46$E-05$	3.23$E-05$	1.17$E-04$	1.73$E-06$	6.86$E-05$	4.88$E-05$	3.59$E-05$	1.91$E-06$	1.23$E-04$	4.33$E-05$	1.55$E-05$
V	5.98$E-07$	5.97$E-05$	2.96$E-05$	7.46$E-05$	1.07$E-06$	6.29$E-05$	4.45$E-05$	2.23$E-05$	1.63$E-06$	1.04$E-04$	3.72$E-05$	9.00$E-06$

Table 19: Convergence analysis of population density of local, immigrant prey, and predator under the influence of variations in β_1.

	Cases	FIT				MAD			TIC			ENSE		
		$\leq 10^{-6}$	$\leq 10^{-7}$	$\leq 10^{-8}$	$\leq 10^{-9}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-7}$
$S(t)$	I	89	55	16	0	99	83	6	100	91	35	71	38	4
	II	87	61	25	6	100	84	10	100	94	29	73	39	7
	III	98	80	27	2	100	79	12	100	94	25	78	37	12
	IV	98	77	25	3	100	85	8	100	93	22	86	43	9
	V	90	84	40	4	100	81	19	100	94	41	83	60	21
$X(t)$	I	97	75	36	11	99	77	15	100	85	21	94	66	27
	II	99	81	48	14	100	83	11	100	90	17	96	69	27
	III	98	84	33	12	100	82	7	100	90	9	96	65	19
	IV	99	84	39	11	100	80	10	100	92	14	71	17	3
	V	77	27	5	0	100	82	8	100	89	16	93	61	32
$Y(t)$	I	97	75	36	11	99	62	3	100	92	23	99	80	41
	II	94	78	36	8	100	68	4	100	90	24	97	79	38
	III	81	37	4	0	99	65	5	100	87	20	80	49	9
	IV	97	74	34	7	99	67	4	100	90	21	81	43	11
	V	77	27	5	0	100	97	42	100	100	66	87	60	27

Four cases are considered, depending on the value of α_2.

Case I: $\alpha_2 = 0.05$

Case II: $\alpha_2 = 0.10$

Case III: $\alpha_2 = 0.15$

Case IV: $\alpha_2 = 0.25$

The LeNN-WOA-NM algorithm is used to optimize the population densities of equation (33). Table 21 represents the comparison between the Ranga–Kutta method and the proposed technique LeNN-WOA-NM. A comparison between population density of local prey $S(t)$, immigrant prey $X(t)$, and predator $Y(t)$ with variations in α_2 is shown in Table 22. Statistics of absolute errors are given in Table 23 and graphically presented in Figure 10. The behavior of the fitness function for different cases is shown on boxplots as demonstrated in Figure 11. The mean values of the fitness function lie around 10^{-7} as shown in Tables 24 and 25. Unknown parameters achieved by the LeNN-WOA-NM algorithm for the given problem are given in Table 26. From solutions (see Figure 12), the following conclusion can be drawn:

(i) Population density of local prey, immigrant prey, and predator varies directly with an increasing rate of immigrate prey α_2

7.4. Problem IV. In this problem, the effect of variations in the negative force of interaction between local prey and immigrant prey on population densities of the prey-predator model is discussed. An error-based fitness function along with initial populations is given as follows:

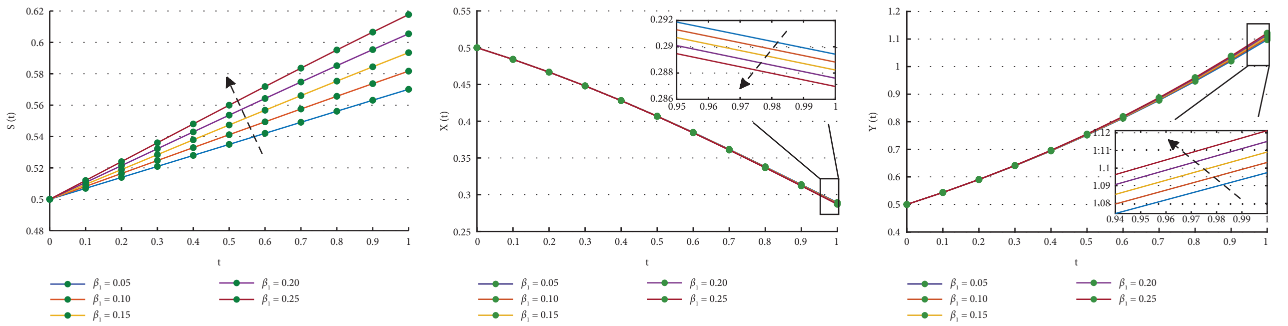
$$\text{Minimize}\, \epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5 + \epsilon_6, \tag{34}$$

where ϵ_1 to ϵ_6 are defined as follows:

$$\begin{cases} \epsilon_1 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{34}\left(\dfrac{dS_n}{dt} - S_n\left((0.12) - \dfrac{(0.12)S_n}{50}\right) - (0.2)S_nX_n + (0.01)S_nY_n\right)^2, \\ \epsilon_2 = \dfrac{1}{N}\displaystyle\sum_{n=35}^{68}\left(\dfrac{dX_n}{dt} - X_n\left((0.2) - \dfrac{(0.2)X_n}{k_2}\right) + \beta_2 S_nX_n + (0.9)X_nY_n\right)^2, \\ \epsilon_3 = \dfrac{1}{N}\displaystyle\sum_{n=69}^{102}\left(\dfrac{dY_n}{dt} - (0.9)S_nY_n - (0.8)X_nY_n + (0.01)Y_n + (0.01)Y_n^2\right)^2, \\ \epsilon_4 = (S_0 - 0.5)^2, \\ \epsilon_5 = (X_0 - 0.5)^2, \\ \epsilon_6 = (Y_0 - 0.5)^2. \end{cases} \tag{35}$$

Table 20: Unknown parameters obtained by the LeNN-WOA-NM algorithm for the prey-predator model under the influence of variations in β_1.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
	1	0.43006915	0.1953118	0.58526278	0.2039733	1.09461748	−0.3042383	0.45137375	0.613472621	0.21300391
	2	−0.1817023	0.12967851	−0.2238481	−0.1299324	−0.2205129	−0.2434037	0.89383923	0.347571185	0.192691905
	3	−0.1639676	−0.1906971	0.01654308	−0.2683275	0.21586942	−0.0650826	0.15826136	−0.38637064	0.544846185
	4	0.6432707	−0.2243964	−0.0644709	−0.1415483	0.00139286	−0.1221445	0.21055624	−0.28460075	0.325283188
	5	−0.063381	0.20479542	−0.0837701	0.37097529	−0.2551801	−0.1457444	0.85822173	−0.35898385	0.095809867
Case I	6	−0.0587215	0.09571579	−0.2151334	0.23857131	0.30661251	0.34850165	−0.005745	0.620231649	0.59934855
	7	0.31197992	−0.0344083	−0.0542888	0.01997741	−0.1641653	−0.0515449	0.00074005	0.000287462	−0.37381232
	8	0.85365924	−0.0169789	0.25572233	0.80492801	−0.0023113	−0.2852425	−0.0522238	−0.2275238	0.146324864
	9	−0.2837938	−0.069528	−0.2221892	0.31692949	0.34731804	−0.0409553	−0.168709	−0.12570174	0.029229593
	10	−0.2720053	−0.0083273	0.71303279	0.0130666	0.04347241	0.10490579	1.16491711	−0.14620103	0.240782136
	11	0.0232457	0.04413909	0.57180799	−0.3961779	0.13805184	−0.0250827	0.1189892	−0.09976743	0.676863876
	1	0.20084257	0.01848105	0.01151112	0.18834185	0.07714385	1.00199221	0.33090711	0.591954522	0.172081801
	2	0.77731512	0.20280783	−0.1534665	−0.0898678	−0.0222164	0.05097659	0.39013928	−0.16168208	−0.87414288
	3	−0.8932687	0.01562423	−0.8790331	−0.8785872	−0.2400978	0.01126939	0.62600687	−0.07469029	0.293200012
	4	−0.1254086	0.08769516	−0.4580886	−0.2608019	−0.0870728	0.23316388	0.49916445	−0.27253148	−0.00841655
	5	0.05410689	−0.0916165	−0.4295043	−0.5942252	−0.0340513	0.03471072	−0.0634013	−0.11877784	−0.77212021
Case II	6	0.22564844	0.11645954	−0.0194417	0.06716894	−0.0067884	−0.2770185	−0.3074844	−0.17361157	−0.02940934
	7	0.25778166	0.02802331	0.32826015	0.0874406	−0.9239147	0.5776072	0.12893563	−0.07494518	0.101314911
	8	−0.6152252	0.08467719	−0.005507	−0.0016885	0.26959676	0.00027392	0.01072972	−0.05967189	0.794011281
	9	−0.4916591	0.03423008	0.35221385	0.1673546	0.20097946	0.03970559	−0.5329031	0.05685715	0.093143062
	10	0.09372071	0.45237283	−0.4270023	0.12270954	0.01633451	−0.0244912	0.32114085	−0.08529571	0.896172613
	11	1.84E − 05	−0.1783898	−0.0967566	−0.032303	−0.3339006	−0.0531862	0.05299049	0.00250836	0.0227459
	1	0.32865927	0.73232241	−0.0426226	0.3680448	0.12592558	−0.1711813	0.6895093	0.365675977	0.1815117
	2	0.06352326	0.01615348	0.71943001	0.02765614	0.5770961	0.12508308	0.72579383	0.234207148	0.278978323
	3	0.05902058	0.74414566	−0.0387953	−0.1285802	0.0259792	0.02360138	−0.0390772	0.334378871	0.373885739
	4	0.01672986	0.30780827	−0.0305187	0.76478835	0.00922025	−0.0003176	0.39416458	−0.14105866	0.231267475
	5	0.16886906	0.16663601	−0.1557756	0.14372478	−0.0053584	0.39867936	0.53867137	0.083576797	−0.0930085
Case III	6	−0.1209806	−0.3749588	0.21620292	0.04682217	0.16582226	−0.1813835	0.31094709	−0.11844016	0.003817292
	7	−0.0756049	0.26868517	0.11232029	0.21554735	0.08172268	0.33302842	0.25825317	0.129740332	−0.27092193
	8	0.17875086	−0.0130043	0.7761683	−0.2683637	−0.0568285	0.08575422	−0.3405182	0.078523738	−0.1038394
	9	0.52179302	0.03458665	0.06475427	0.06956231	0.25072313	0.02715948	0.25907261	0.038549325	−0.02300298
	10	0.44795101	0.14647434	0.16811506	−0.0008598	0.31689128	0.21717058	0.55924027	0.030536512	−0.63332453
	11	0.07886187	−0.1241851	0.13943947	0.01341213	−0.6267738	0.34912288	−0.1931707	−0.01298631	0.579458203
	1	0.36926007	0.34021884	−0.0935731	0.16169213	0.09246816	1.38553805	0.44058544	0.499926759	0.127460796
	2	0.32481477	0.21004215	0.17360292	0.06611299	−0.152444	0.17230255	0.4871752	0.240390062	−0.13247269
	3	−0.1570069	0.04398714	−0.1372116	0.10393927	0.27712388	0.21929305	0.35934559	0.072436543	0.599680184
	4	0.10042725	−0.0405233	−0.0276124	0.00768115	−0.1152797	0.46671001	0.32697635	0.218813803	0.11578542
	5	0.2981599	0.07895408	0.05929431	−0.2575791	0.18210718	0.27949753	0.11208642	−0.13550479	−0.19303355
Case IV	6	0.01774278	0.0794438	0.34301322	0.3079287	−0.0288429	0.35285173	0.08051491	−0.0383289	0.301524551
	7	0.15448802	−0.0290092	0.37921177	0.40577398	0.02317734	0.19760018	0.5453665	0.190128144	0.134972259
	8	0.07303209	0.08335033	−0.0272924	−0.0268407	0.11495061	0.17565862	0.14548013	0.050459546	0.999559566
	9	0.06638016	0.09365192	0.46046701	0.31157149	0.24788627	0.39745699	0.14745623	−0.0203899	0.11010541
	10	0.14574411	0.27019012	0.08964107	−0.013653	0.6999887	0.14706426	0.14311693	0.085587473	0.406653134
	11	0.00422723	−0.0202326	0.09754264	4.84E − 05	0.01130254	0.4544351	0.20276643	−0.02176553	0.041899718
	1	0.41668155	0.55141417	0.31705647	0.53356928	−2.0883662	−0.2610704	0.24747335	−0.49568348	0.15116601
	2	0.28111104	0.29873199	0.12415861	0.01035183	0.3912644	−0.2250856	0.11463835	0.572594198	0.639529712
	3	−0.2880051	0.47632466	0.02148248	0.42831754	0.25753792	−0.1067969	0.49570912	−0.56574524	0.140211616
	4	0.28382154	0.07492232	0.25762031	−0.0051616	0.449995	−0.2583676	0.00862342	0.042826998	−0.24532638
	5	−0.1828993	0.05610035	0.41157899	0.10823296	−0.0022137	0.43840826	0.14531787	0.504082565	0.073357981
Case V	6	−0.3536545	0.4045564	0.05410188	−0.1039612	−0.4487177	0.18545137	0.00207541	0.071838234	−0.18198509
	7	0.23523025	−0.1197378	0.22737042	0.27683837	−0.4502542	0.53741643	−0.1929315	0.094579762	0.334936445
	8	−0.0101984	0.60796793	0.06730913	−0.0740784	0.1513789	0.63740825	−0.3265244	−0.27219855	0.235174666
	9	−0.0038892	−0.3239558	0.10405389	0.41194902	0.01303423	−0.058557	0.25579472	0.192432911	−0.11924924
	10	0.20983244	0.24683053	0.09135039	0.0475623	0.44987648	−0.0005977	−0.2977078	0.013114946	0.599953402
	11	0.17699185	0.19078854	−0.4635923	−0.0023057	−0.2080841	0.13883293	0.21279433	−0.19995201	0.280170884



(a) (b) (c)

FIGURE 9: Comparison between solutions obtained by the LeNN-WOA-NM algorithm under the influence of variations in β_1 on population densities of the prey-predator model. (a) Approximate solutions for population density of local prey. (b) Approximate solutions for population density of immigrant prey. (c) Approximate solutions for population density of the predator.

TABLE 21: Comparison between solutions obtained by the LeNN-WOA-NM algorithm and ode45 for the prey-predator model with $\alpha_2 = 0.15$.

t	$S(t)$	ode45	$X(t)$	ode45	$Y(t)$	ode45
0.0	0.50000060	0.5	0.49999713	0.5	0.50000085	0.5
0.2	0.52140043	0.5214836	0.46218751	0.46914875	0.59021206	0.59055511
0.4	0.54273869	0.54304303	0.41967719	0.43242894	0.69476821	0.69630534
0.6	0.56382241	0.56446804	0.37330484	0.39042402	0.81497365	0.81880710
0.8	0.58443923	0.58554139	0.32433283	0.34420213	0.95201479	0.95948647
1.0	0.60442948	0.60605527	0.27440198	0.29530956	1.10702186	1.11963769

Three cases are considered, depending on the value of β_2.

Case I: $\beta_2 = 0.05$

Case II: $\beta_2 = 0.10$

Case III: $\beta_2 = 0.20$

The LeNN-WOA-NM algorithm is used to optimize the population densities of prey-predator model equation (35). Population densities of local prey, immigrant prey, and the predator are given in Tables 27-28, while absolute errors are shown in Table 29. The absolute errors in the solutions of the proposed technique lie between 10^{-9} and 10^{-12} as shown in Figure 13. Table 30 shows global values of performance indices and convergence of the fitness value as shown in Figure 14. The statistics shown in Table 31 illustrate that the global values of different performance functions lie between 10^{-5} and 10^{-7}, which highlights the robustness of the technique. The values of the weights in the LeNN structure for recreation of the approximate solutions are given in Table 32. From Figure 15, the following conclusion can be drawn:

(i) Population density of local prey varies directly with variations in β_2

(ii) Population density of immigrant prey and predator varies inversely with variations in β_2

7.5. Problem V: Effect of Variation in γ_1 on the Prey-Predator Model. In this problem, the effect of variations in the catching rate of local prey γ_1 on population densities of the prey-predator model is discussed. An error-based fitness function along with initial populations is given as follows:

$$\text{Minimize}\ \varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6, \tag{36}$$

where ϵ_1 to ϵ_6 are defined as follows:

$$\begin{cases} \epsilon_1 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{34}\left(\dfrac{\mathrm{d}S_n}{\mathrm{d}t} - S_n\left((0.12) - \dfrac{(0.12)S_n}{50}\right) - (0.2)S_nX_n + \gamma_1 S_nY_n\right)^2, \\ \epsilon_2 = \dfrac{1}{N}\displaystyle\sum_{n=35}^{68}\left(\dfrac{\mathrm{d}X_n}{\mathrm{d}t} - X_n\left((0.2) - \dfrac{(0.2)X_n}{k_2}\right) + (0.1)S_nX_n + (0.9)X_nY_n\right)^2, \\ \epsilon_3 = \dfrac{1}{N}\displaystyle\sum_{n=69}^{102}\left(\dfrac{\mathrm{d}Y_n}{\mathrm{d}t} - (0.9)S_nY_n - (0.8)X_nY_n + (0.01)Y_n + (0.01)Y_n^2\right)^2, \\ \epsilon_4 = (S_0 - 0.5)^2, \\ \epsilon_5 = (X_0 - 0.5)^2, \\ \epsilon_6 = (Y_0 - 0.5)^2. \end{cases} \tag{37}$$

Table 22: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population densities of local, immigrant prey, and predator under the influence of the increasing rate of immigrant prey α_2.

t	$S(t)$				$X(t)$				$Y(t)$			
	$\alpha_2 = 0.05$	$\alpha_2 = 0.10$	$\alpha_2 = 0.15$	$\alpha_2 = 0.25$	$\alpha_2 = 0.05$	$\alpha_2 = 0.10$	$\alpha_2 = 0.15$	$\alpha_2 = 0.25$	$\alpha_2 = 0.05$	$\alpha_2 = 0.10$	$\alpha_2 = 0.15$	$\alpha_2 = 0.25$
0.0	0.49999996	0.49999970	0.50000060	0.50000018	0.49999997	0.50000023	0.49999713	0.49999870	0.499999997	0.500000562	0.500000847	0.500001033
0.2	0.52131291	0.52135689	0.52140043	0.52150112	0.45321205	0.45768261	0.46218751	0.47137924	0.589770367	0.589997848	0.590212055	0.590665856
0.4	0.54235797	0.54255447	0.54273869	0.54312436	0.40356843	0.41154623	0.41967719	0.43644215	0.692819675	0.693794106	0.694768206	0.696794465
0.6	0.56299694	0.56341623	0.56382241	0.56466616	0.35216917	0.36258531	0.37330484	0.39568861	0.810159712	0.812546426	0.814973647	0.820021754
0.8	0.58308161	0.58375622	0.58443923	0.58587662	0.30037587	0.31213132	0.32433283	0.35016888	0.942778570	0.947340120	0.952014792	0.961823425
1.0	0.60245464	0.60342550	0.60442948	0.60653682	0.24968763	0.26176158	0.27440198	0.30144245	1.091704823	1.099236698	1.107021862	1.123532428

Table 23: Absolute errors obtained by the LeNN-WOA-NM algorithm for population densities of local, immigrant prey, and predator under the influence of α_2.

t	$S(t)$				$X(t)$				$Y(t)$			
	$\alpha_2 = 0.05$	$\alpha_2 = 0.10$	$\alpha_2 = 0.15$	$\alpha_2 = 0.25$	$\alpha_2 = 0.05$	$\alpha_2 = 0.10$	$\alpha_2 = 0.15$	$\alpha_2 = 0.25$	$\alpha_2 = 0.05$	$\alpha_2 = 0.10$	$\alpha_2 = 0.15$	$\alpha_2 = 0.25$
0	$1.38E-08$	$1.54E-09$	$2.74E-09$	$1.77E-09$	$1.18E-09$	$1.88E-10$	$4.68E-09$	$6.17E-09$	$3.17E-10$	$6.73E-11$	$8.97E-10$	$2.19E-10$
0.2	$6.09E-09$	$1.40E-09$	$3.92E-09$	$3.57E-09$	$6.24E-10$	$1.22E-10$	$6.51E-10$	$9.47E-09$	$1.15E-11$	$4.00E-11$	$9.21E-10$	$7.22E-11$
0.4	$4.30E-09$	$1.63E-10$	$1.22E-09$	$3.29E-10$	$1.59E-09$	$8.95E-11$	$5.03E-09$	$1.78E-11$	$7.33E-10$	$1.29E-10$	$1.48E-10$	$4.81E-10$
0.6	$2.00E-09$	$6.47E-10$	$2.54E-10$	$1.89E-09$	$5.33E-10$	$1.49E-11$	$2.95E-09$	$8.36E-10$	$1.06E-09$	$9.40E-12$	$5.36E-10$	$5.34E-10$
0.8	$8.76E-09$	$7.02E-10$	$1.16E-09$	$3.77E-09$	$9.98E-11$	$9.28E-11$	$7.06E-10$	$3.10E-11$	$1.31E-10$	$2.41E-11$	$2.99E-10$	$4.31E-10$
1	$1.40E-08$	$9.57E-10$	$1.35E-09$	$6.00E-09$	$3.71E-10$	$5.62E-12$	$5.19E-09$	$1.12E-11$	$7.18E-10$	$9.95E-12$	$4.08E-10$	$3.35E-10$

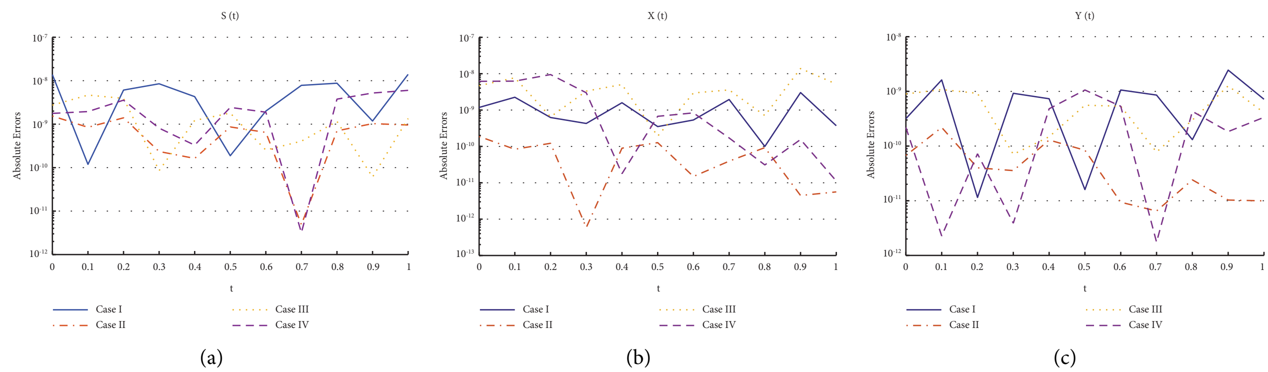


Figure 10: Absolute errors obtained by the LeNN-WOA-NM algorithm under the influence of variations in α_2 on population densities of the prey-predator model. (a) Population density of local prey. (b) Population density of immigrant prey. (c) Population density of the predator.

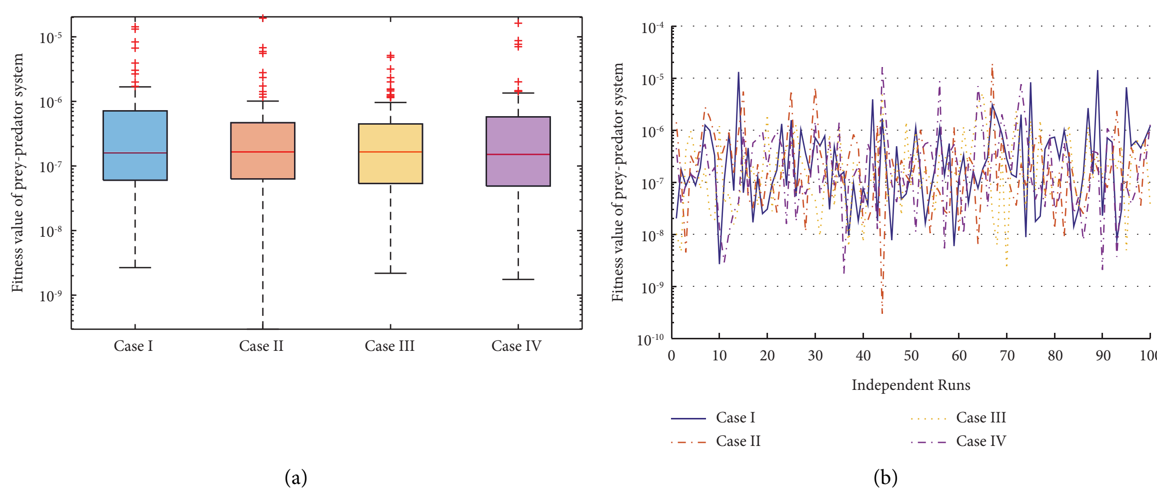


Figure 11: Comparison between the box plots and convergence graphs of fitness evaluation over 100 independent runs for the prey-predator model with variations in the increasing rate of immigrant prey. (a) Box plots for fitness evaluation of the prey-predator model with variations in α_2. (b) Convergence of the fitness value of the prey-predator model with variations in α_2.

Four cases are considered, depending on variations in γ_1.

Case I: $\gamma_1 = 0.005$

Case II: $\gamma_1 = 0.015$

Case III: $\gamma_1 = 0.020$

Case IV: $\gamma_1 = 0.025$

The LeNN-WOA-NM algorithm is used to optimize the population densities of equation (37). Table 33 represents the comparison between the Ranga–Kutta method (ode45) and the proposed technique LeNN-WOA-NM algorithm. Approximate solutions for population density of local prey $S(t)$, immigrant prey $X(t)$, and the predator $Y(t)$ with different values of γ_1 are presented in Table 34. Statistics of absolute errors are given in Table 35 and graphically shown in Figure 16. The minimum and mean values of the fitness function for each case lie between 10^{-9} and 10^{-7}, respectively as illustrated in Figure 17. Convergence analysis of the fitness value, MAD, TIC, and ENSE is given in Tables 36-37. Unknown neurons of LeNN are shown in Table 38. From Figure 18, the following conclusion can be drawn:

(i) Population density of local prey and population density of the predator vary inversely with variations in γ_1

(ii) Population density of immigrant prey has a direct relation with variations in γ_1

7.6. Problem VI: Effect of Variations in γ_2 on the Prey-Predator Model. In this problem, the effect of variations in the catching rate of immigrant prey on population densities of

Table 24: Statistics of global performance indices for variations in the intrinsic growth rate of immigrant prey on the prey-predator model.

Cases	$S(t)$				$X(t)$				$Y(t)$			
	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE
I	$2.25E-06$	$5.70E-05$	$2.85E-05$	$7.97E-05$	$9.91E-07$	$6.19E-05$	$4.75E-05$	$7.97E-05$	$6.03E-07$	$8.42E-05$	$3.00E-05$	$5.97E-06$
II	$1.21E-06$	$4.64E-05$	$2.36E-05$	$4.09E-05$	$5.08E-07$	$4.68E-05$	$3.50E-05$	$8.22E-06$	$4.05E-07$	$6.93E-05$	$2.49E-05$	$2.93E-06$
III	$1.24E-06$	$4.33E-05$	$2.17E-05$	$3.13E-05$	$4.31E-07$	$4.76E-05$	$3.46E-05$	$8.99E-06$	$5.75E-07$	$7.72E-05$	$2.82E-05$	$3.20E-06$
IV	$5.22E-07$	$4.55E-05$	$2.29E-05$	$4.13E-05$	$3.65E-07$	$4.91E-05$	$3.39E-05$	$1.15E-05$	$8.39E-07$	$8.64E-05$	$3.06E-05$	$4.58E-06$

Table 25: Convergence analysis of population density of local prey, immigrant prey, and predator under the influence of variations in the increasing rate of immigrant prey α_2.

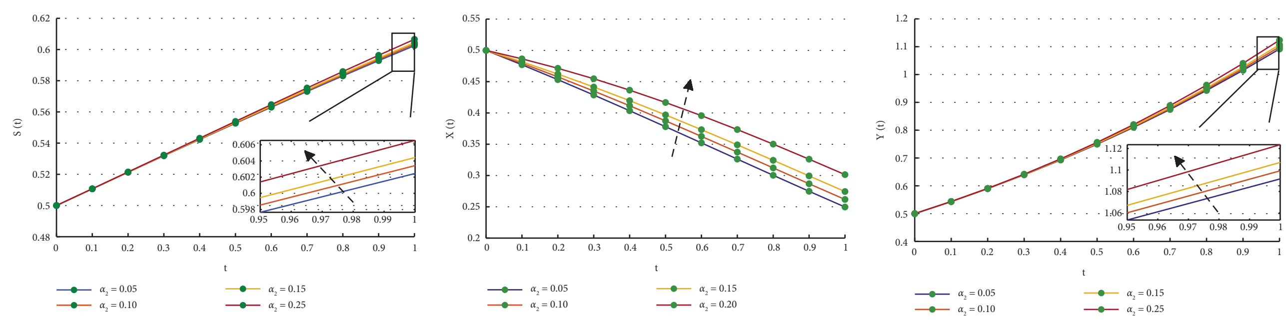
	Cases	FIT				MAD			TIC			ENSE		
		$\le 10^{-6}$	$\le 10^{-7}$	$\le 10^{-8}$	$\le 10^{-9}$	$\le 10^{-4}$	$\le 10^{-5}$	$\le 10^{-6}$	$\le 10^{-4}$	$\le 10^{-5}$	$\le 10^{-6}$	$\le 10^{-5}$	$\le 10^{-6}$	$\le 10^{-7}$
$S(t)$	I	94	62	20	1	100	89	9	100	96	29	89	46	9
	II	100	70	16	1	100	91	14	100	98	26	91	42	14
	III	99	65	16	1	100	96	11	100	99	27	96	47	11
	IV	100	87	38	6	100	85	20	100	98	42	87	53	21
$X(t)$	I	98	78	45	14	100	85	13	100	90	17	95	65	37
	II	100	89	54	17	100	87	11	100	92	19	100	86	38
	III	99	92	52	11	100	89	8	100	97	17	99	79	32
	IV	100	89	46	16	100	87	12	100	96	21	98	71	27
$Y(t)$	I	99	86	39	12	100	76	5	100	95	27	97	89	72
	II	100	92	46	11	100	79	6	100	97	31	100	93	60
	III	100	83	46	15	100	74	8	100	99	26	100	87	56
	IV	98	78	43	11	100	69	4	100	98	19	99	93	54

Table 26: Unknown parameters obtained by the LeNN-WOA-NM algorithm for the prey-predator model under the influence of variations in the increasing rate of immigrant prey α_2.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
Case I	1	0.29809866	−0.0315366	0.58012735	0.47297503	0.16870771	0.25422827	0.04487536	0.494241827	0.219042842
	2	0.07902509	−0.142623	−0.5326572	0.29933207	0.36334065	0.40051667	0.7993046	0.01500143	0.046498829
	3	0.37716273	−0.0698453	−0.5278093	−0.5919778	0.06003798	−0.6780052	−0.608852	0.609110992	0.116343672
	4	0.06418686	0.24209073	0.11910483	0.29508708	−0.0084275	0.0309784	−0.0262412	−0.34143893	−0.53664746
	5	−0.0499254	−0.0250403	0.05106686	0.06326291	0.38462339	0.07701417	0.12151676	0.119098177	0.09960376
	6	−0.0395958	0.01489951	−0.2883484	0.08480605	0.18102453	−0.0291467	0.00187112	−0.01144597	−0.34122987
	7	0.15684755	0.08609064	0.20735343	0.51895958	0.4033704	0.21052592	0.49856342	0.262653738	0.034559337
	8	−0.0340164	0.04863621	0.22276639	0.04403982	0.21344694	0.31828637	0.24286503	0.043774589	0.529564847
	9	−0.2519217	0.061991	0.13654615	0.1880865	0.14486401	0.37615985	0.46832364	0.174620368	0.266799487
	10	0.54249116	−0.0324119	0.22565323	0.10730714	0.21764342	0.30007676	0.07731738	0.211031089	0.199472911
	11	0.12904085	0.04855899	−0.4666845	0.25677349	−0.0420107	−0.1514589	−0.1778985	0.104252082	−0.3576282
Case II	1	0.22014881	−0.0955389	0.01922733	0.44099269	0.16902098	0.21058167	0.56662276	0.214534099	−0.12547843
	2	−0.510556	0.02522092	0.56596508	0.2853307	−0.0140499	0.1381917	−0.0004646	0.491829768	0.382367631
	3	−0.1657635	0.01472461	−0.1147983	0.39351912	0.21069293	−0.2685875	0.39509887	0.072673362	−0.18283919
	4	0.05282327	0.07711052	−0.2051438	0.12992756	0.19352914	0.2499045	0.43645858	0.382113573	0.058714001
	5	0.64499253	0.25762641	0.23808203	0.8495944	−0.0043216	−0.0017105	0.25457162	0.313158881	−0.13553561
	6	0.09858966	−0.0273145	0.48787987	0.09374791	0.21889407	−0.0062504	−0.1344748	0.044741355	−0.2149526
	7	0.23380371	−0.1260215	0.3289594	0.05799592	−0.333364	0.15201786	−0.4456943	−0.06976927	0.002138263
	8	0.16035874	−0.1136321	−0.0036501	0.01897557	−0.0650362	0.09091915	1.14914925	−0.03585999	0.165910127
	9	0.01683166	0.03027356	0.01366012	0.16775817	0.20057677	0.1459576	0.53828172	0.130328528	0.267951969
	10	0.26771936	0.12204652	0.2996044	−0.1152812	0.18819176	0.17111444	−0.3869172	0.095372589	−1.04663455
	11	−0.1430149	0.0042418	−0.1108715	−0.1402711	0.17676553	−0.2193704	−0.2747091	0.12544106	0.108988227
Case III	1	0.20833477	0.87977164	−0.329864	0.37497171	0.15803512	1.08685499	0.81716797	−0.8285298	0.207619724
	2	−0.009672	−0.0834121	−0.0062989	0.00506333	0.45907436	−0.2830568	−0.055188	−1.29640697	0.383186286
	3	−0.1260443	0.5466368	−0.2547923	−0.1978145	−0.3014792	0.01482453	0.16026646	−0.12624417	−0.14211688
	4	−0.0459971	−0.1850824	0.44010074	0.85262434	−0.2293558	−0.1803312	0.15934035	−0.27875907	0.925897653
	5	0.61609669	−0.0654918	−0.2249982	1.06003476	−0.0605608	0.0377206	−0.2465468	0.095037929	−0.00368046
	6	−0.4221877	0.1915771	0.29040894	−0.1447329	−0.1716272	−0.2814112	0.10252045	0.140572643	0.100666708
	7	0.12304762	−0.0412171	−0.2947437	−0.3344259	0.15047463	−0.0948965	−0.3952389	−0.18667499	0.22033229
	8	0.22471377	−0.1369576	−0.0135929	0.46118375	0.09675102	0.25499179	0.6880506	−0.04146236	−0.02009528
	9	−0.0028525	0.28384483	−0.0137988	0.86855616	0.15119596	0.43141743	−0.310849	0.177047029	0.639644264
	10	0.10012064	−0.0365188	−0.3495845	0.3752446	0.15082062	0.39737423	0.02610775	−0.11632104	0.264941005
	11	0.6577301	−0.1729186	0.28691512	−0.277895	−0.2620078	−0.3183686	0.4666604	0.182729564	−0.40213292

Table 26: Continued.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
Case IV	1	0.27584273	0.14440132	0.05159123	−0.0014501	0.02944933	0.43857897	0.41541694	0.364106607	0.137637467
	2	1.0378582	0.22634893	−0.1336449	0.37479484	0.32390138	0.11683191	1.10634191	0.743870281	0.133721081
	3	0.30555234	0.14178372	0.09870863	0.1378461	−0.0134231	0.25235605	0.15847534	0.546747961	0.267557848
	4	0.49396717	−0.0077854	0.12863956	0.3091184	−0.0434135	0.98475694	0.07222518	−0.0015274	0.03284215
	5	0.96461172	0.48761051	0.0267945	0.4431833	0.19500184	0.28889858	0.10389701	−0.06048102	0.012779125
	6	−0.0311005	−0.2458898	0.01323704	0.39215778	0.07525855	0.0342729	−0.0239009	0.305047682	0.240819746
	7	−0.0009785	0.27803419	0.43494014	0.0813522	0.16527361	0.10128603	−0.0235703	0.131236783	0.342597408
	8	0.03245948	0.05591694	0.9180906	−0.064116	0.19286301	0.46309077	−0.0226751	0.265693128	0.191819022
	9	0.04135314	−0.0595634	−0.018826	−0.0453863	0.10473006	0.09188733	0.12139352	0.281687652	0.039757676
	10	0.06206464	0.10871955	0.21480649	0.12965376	−0.0275439	0.21169344	0.08952384	−0.23209362	−0.11227638
	11	0.41183634	−0.0123283	0.31935826	−0.0629837	0.00702969	0.08486244	0.22578589	0.194356561	0.207691077



(a) (b) (c)

Figure 12: Comparison between solutions obtained by the LeNN-WOA-NM algorithm under the influence of variations in the increasing rate of immigrant prey on population densities of the prey-predator model. (a) Approximate solutions for population density of local prey. (b) Approximate solutions for population density of immigrant prey. (c) Approximate solutions for population density of the predator.

Table 27: Comparison between solutions obtained by the LeNN-WOA-NM algorithm and ode45 for the prey-predator model under the influence of variations in β_2.

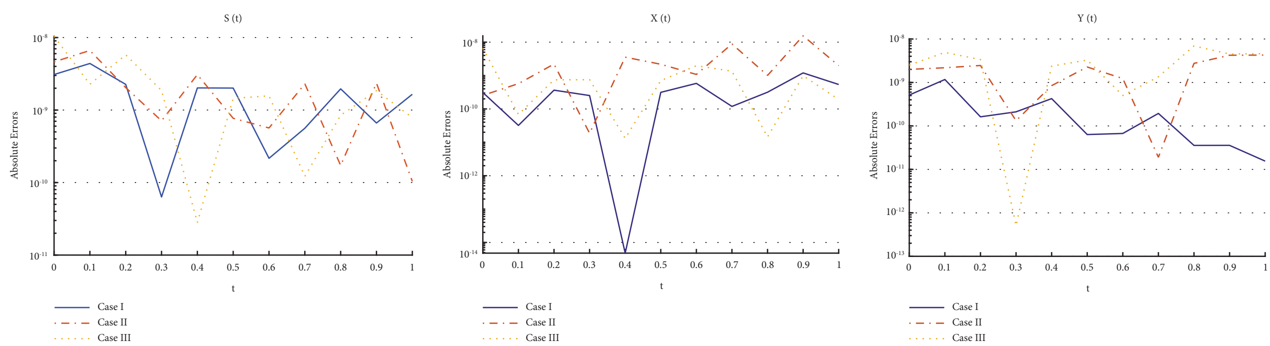
t	$S(t)$	*ode*45	$X(t)$	*ode*45	$Y(t)$	*ode*45
0.0	0.50000260	0.5	0.49999663	0.5	0.50000434	0.5
0.2	0.52140524	0.52140803	0.46202665	0.46202728	0.59020495	0.59020976
0.4	0.54273086	0.54274216	0.41918897	0.41919322	0.69472629	0.69473392
0.6	0.56379821	0.56380333	0.37243267	0.37243138	0.81483980	0.81483944
0.8	0.58439482	0.58439812	0.32308553	0.32307837	0.95168295	0.95168948
1.0	0.60434466	0.60435385	0.27281091	0.27280834	1.10636629	1.10638491

Table 28: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population densities of local prey, immigrant prey, and predator under the influence of the negative impact of the force of interaction on immigrant prey β_2.

t	$S(t)$			$X(t)$			$Y(t)$		
	$\beta_2 = 0.05$	$\beta_2 = 0.10$	$\beta_2 = 0.20$	$\beta_2 = 0.05$	$\beta_2 = 0.10$	$\beta_2 = 0.20$	$\beta_2 = 0.05$	$\beta_2 = 0.10$	$\beta_2 = 0.20$
0.0	0.49999997	0.49999839	0.50000260	0.49999998	0.50000038	0.49999663	0.49999998	0.49999971	0.500004344
0.2	0.52147482	0.52146698	0.52140524	0.46914810	0.46676681	0.46202665	0.59055020	0.59044621	0.590204953
0.4	0.54303474	0.54294731	0.54273086	0.43242621	0.42797254	0.41918897	0.69630031	0.69578968	0.694726289
0.6	0.56446751	0.56424461	0.56379821	0.39042495	0.38432545	0.37243267	0.81880214	0.81747363	0.814839800
0.8	0.58553673	0.58516310	0.58439482	0.34420481	0.33701910	0.32308553	0.95947828	0.95685274	0.951682945
1.0	0.60604682	0.60548065	0.60434466	0.29530830	0.28760242	0.27281091	1.11962732	1.11514838	1.106366286

Table 29: Absolute errors obtained by the LeNN-WOA-NM algorithm for population densities of local prey, immigrant prey, and predator under the influence of the negative impact of the force of interaction on immigrant prey β_2.

t	$S(t)$			$X(t)$			$Y(t)$		
	$\beta_2 = 0.05$	$\beta_2 = 0.10$	$\beta_2 = 0.20$	$\beta_2 = 0.05$	$\beta_2 = 0.10$	$\beta_2 = 0.20$	$\beta_2 = 0.05$	$\beta_2 = 0.10$	$\beta_2 = 0.20$
0.0	$3.09E-09$	$4.62E-09$	$1.04E-08$	$3.11E-10$	$2.55E-10$	$6.59E-09$	$5.14E-10$	$2.00E-09$	$2.51E-09$
0.2	$2.26E-09$	$2.04E-09$	$5.78E-09$	$3.65E-10$	$2.21E-09$	$7.44E-10$	$1.62E-10$	$2.46E-09$	$3.36E-09$
0.4	$2.02E-09$	$3.08E-09$	$2.81E-11$	$4.80E-15$	$3.55E-09$	$1.32E-11$	$4.25E-10$	$8.45E-10$	$2.38E-09$
0.6	$2.16E-10$	$5.66E-10$	$1.57E-09$	$5.83E-10$	$1.08E-09$	$1.97E-09$	$6.71E-11$	$1.23E-09$	$5.01E-10$
0.8	$1.96E-09$	$1.72E-10$	$8.46E-10$	$3.17E-10$	$1.01E-09$	$1.45E-11$	$3.56E-11$	$2.75E-09$	$6.92E-09$
1.0	$1.65E-09$	$1.04E-10$	$7.87E-10$	$5.40E-10$	$1.98E-09$	$1.91E-10$	$1.55E-11$	$4.27E-09$	$4.61E-09$



(a) (b) (c)

Figure 13: Absolute errors obtained by the LeNN-WOA-NM algorithm under the influence of variations in β_2 on population densities of the prey-predator model. (a) Population density of local prey. (b) Population density of immigrant prey. (c) Population density of the predator.

the prey-predator model is discussed. An error-based fitness function along with initial populations is given as follows:

$$\text{Minimize}\, \varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6. \quad (38)$$

where ϵ_1 to ϵ_6 are defined as follows:

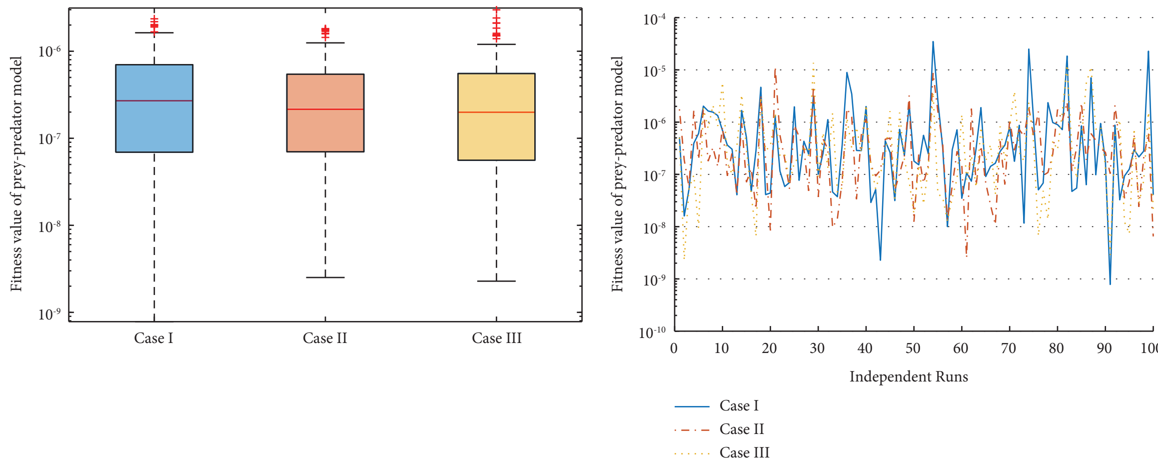
$$\begin{cases} \epsilon_1 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{34}\left(\dfrac{dS_n}{dt} - S_n\left((0.12) - \dfrac{(0.12)S_n}{50}\right) - (0.2)S_nX_n + (0.01)S_nY_n\right)^2, \\ \epsilon_2 = \dfrac{1}{N}\displaystyle\sum_{n=35}^{68}\left(\dfrac{dX_n}{dt} - X_n\left((0.2) - \dfrac{(0.2)X_n}{k_2}\right) + (0.1)S_nX_n + \gamma_2X_nY_n\right)^2, \\ \epsilon_3 = \dfrac{1}{N}\displaystyle\sum_{n=69}^{102}\left(\dfrac{dY_n}{dt} - (0.9)S_nY_n - (0.8)X_nY_n + (0.01)Y_n + (0.01)Y_n^2\right)^2, \\ \epsilon_4 = (S_0 - 0.5)^2, \\ \epsilon_5 = (X_0 - 0.5)^2, \\ \epsilon_6 = (Y_0 - 0.5)^2. \end{cases} \quad (39)$$

Five cases are considered, depending on the value of γ_2.

Case I: $\gamma_2 = 0.7$

Table 30: Statistics of global performance indices for variations in the increasing rate of immigrant prey on the prey-predator model.

Cases	$S(t)$				$X(t)$				$Y(t)$			
	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE
I	$5.58E-07$	$4.70E-05$	$2.36E-05$	$3.69E-05$	$4.72E-07$	$5.09E-05$	$3.59E-05$	$1.10E-05$	$8.00E-07$	$9.48E-05$	$3.40E-05$	$4.71E-06$
II	$1.48E-06$	$5.02E-05$	$2.49E-05$	$5.05E-05$	$4.76E-07$	$5.11E-05$	$3.63E-05$	$1.24E-05$	$7.09E-07$	$9.42E-05$	$3.35E-05$	$5.21E-06$
III	$2.13E-06$	$5.09E-05$	$2.58E-05$	$6.78E-05$	$8.11E-07$	$5.28E-05$	$3.85E-05$	$1.18E-05$	$7.99E-07$	$7.69E-05$	$2.79E-05$	$3.93E-06$



(a) (b)

Figure 14: Comparison between the box plots and convergence graphs of fitness evaluation over 100 independent runs for the prey-predator model with variations in β_2. (a) Box plots for fitness evaluation of the prey-predator model with variations in β_2. (b) Convergence of the fitness value of the prey-predator model with variations in β_2.

Table 31: Convergence analysis of population density of local, immigrant prey, and predator under the influence of variations in β_2.

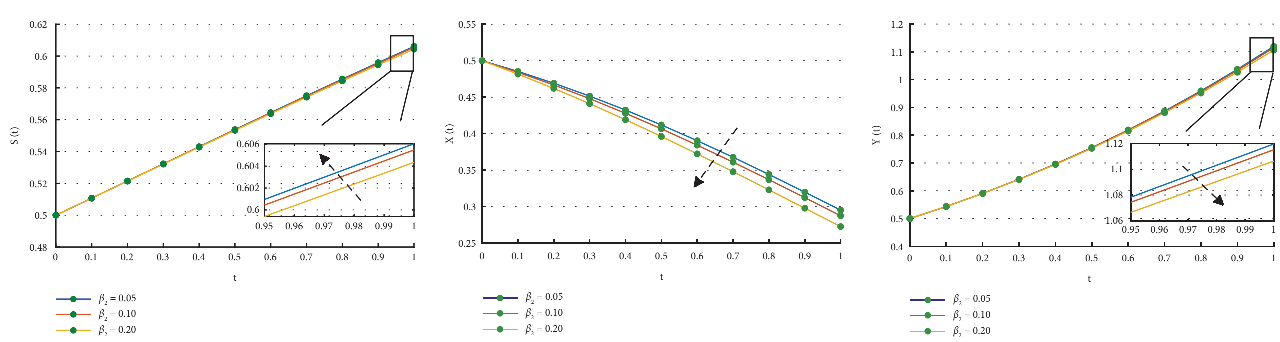
Cases		FIT				MAD			TIC			ENSE		
		$\leq 10^{-6}$	$\leq 10^{-7}$	$\leq 10^{-8}$	$\leq 10^{-9}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-7}$
$S(t)$	I	100	86	32	3	100	92	10	100	100	25	92	44	10
	II	100	63	12	0	100	89	10	100	99	26	89	48	10
	III	96	65	14	2	100	89	15	100	97	37	89	52	15
$X(t)$	I	100	86	45	14	100	88	9	100	94	15	100	68	25
	II	100	91	48	12	100	90	7	100	94	19	97	73	30
	III	99	82	42	15	100	88	12	100	96	20	98	69	27
$Y(t)$	I	99	80	39	5	100	67	2	100	94	24	100	86	48
	II	100	76	39	14	100	71	3	100	93	23	100	88	42
	III	98	86	48	12	100	77	3	100	96	26	100	89	64

Table 32: Unknown parameters obtained by the LeNN-WOA-NM algorithm for the prey-predator model under the influence of variations in β_2.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
Case I	1	0.27535144	0.31269649	−0.212623	0.35537988	0.47248103	0.10822633	0.52114951	0.185720875	−0.00890278
	2	0.61516838	0.24930343	−0.1133566	−0.2698898	0.23075745	1.45662131	0.00285471	0.54326608	−0.03948103
	3	−0.3331712	0.13683002	0.07415358	−0.007558	−0.1414522	0.00429373	−0.2363309	−0.53264189	−0.12625219
	4	−0.1724685	−0.1569281	−0.1780917	−0.0230189	0.06567477	−0.1619849	−0.07434	0.280494463	0.724308841
	5	0.34235255	0.16601501	0.33779612	0.38214222	0.17206398	0.37567544	0.25394552	−0.20424256	0.497829902
	6	0.00647555	−0.0671007	0.41848498	0.465563	0.17852399	0.25910001	0.62807088	−0.25015769	−0.03607794
	7	0.39917241	−0.2086423	0.27916279	−0.1982513	−0.1003214	0.08612276	0.23112207	−0.11265536	−0.04638131
	8	0.37891578	−0.1471092	0.25516272	0.47437651	−0.0479726	0.3954123	0.23371466	0.155785232	0.111525519
	9	−0.1569822	0.21029864	−0.280122	−0.0498625	0.07898872	0.18498889	−0.1404023	−0.28974674	0.273393228
	10	−0.1405838	−0.0119767	−0.1635125	0.11249313	−0.0654565	0.69188379	−0.0877791	−0.1594545	−0.77121428
	11	0.39296939	0.02457412	−0.0200821	−0.1201447	0.17161889	0.03630063	0.04811392	0.279716864	−0.22688875

Table 32: Continued.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
Case II	1	0.50105252	0.82536339	0.58045483	0.45518044	−0.1160988	−0.3093088	0.14735255	0.576552359	0.192127271
	2	0.26141806	0.21796238	0.31815591	−0.1676624	0.4234933	−0.1614025	0.00975838	0.53917788	0.775847618
	3	−0.1877622	−0.1568166	−0.3859022	0.34192011	−0.1162183	0.78757311	0.40990792	−0.29126952	0.094655869
	4	0.02599785	−0.1764539	0.17491514	0.03149905	0.44911473	−0.1695519	−0.0583248	−0.00447879	−0.06289373
	5	−0.2923416	−0.1065739	0.38764839	0.10732316	−0.003522	0.43009606	0.20268435	0.627584449	0.02333884
	6	−0.3868009	0.44970016	−0.0422409	−0.2581683	−0.4105199	0.01656808	0.00066793	0.075964394	−0.18685644
	7	0.05739953	−0.1193984	−0.2607123	0.17181358	−0.2713628	0.66311474	−0.124996	−0.07981308	0.505938132
	8	−0.0216328	0.8067003	0.31701813	−0.1511588	0.14092788	0.65770582	−0.1918301	−0.00407396	0.121012726
	9	$-8.30E-07$	−0.0664668	−0.1168664	0.63180611	−0.1478398	−0.1592677	0.27026843	0.113142772	0.003053122
	10	0.09718671	−0.0187463	−0.0760148	−0.0444081	0.29980599	0.00951493	−0.1584802	−0.01177252	0.821398692
	11	0.19396717	0.18738185	−0.4407164	−0.1400116	−0.1519869	0.03850045	0.32211631	−0.10616995	0.295182432
Case III	1	0.36877614	0.4187033	0.00551721	0.37106378	0.76553227	−0.1135028	0.36766284	0.656543877	0.074012878
	2	0.57123158	0.39029235	0.26925195	0.02897603	−0.0177996	−0.2724976	0.68341531	−0.56659891	0.035271905
	3	−0.2786739	−0.1350866	−0.4079824	0.18172611	−0.1092217	−0.6535241	0.19817369	−0.18104441	0.127718735
	4	0.24896052	0.13322324	0.34245899	0.18535487	0.18336655	0.32712799	−0.1422226	0.152834288	0.025045362
	5	0.29175967	0.21119283	−0.1957816	0.31132875	−0.0532529	−0.3305964	0.04517261	−0.73533086	−0.21051239
	6	0.38609249	0.27590847	0.58209501	−0.535581	−0.1823894	0.31125028	−0.0033038	0.082666375	0.323016755
	7	0.01164667	−0.1421768	0.24328317	0.26480037	0.02173502	0.07848485	0.61775482	0.231893374	0.074346878
	8	0.59381321	0.06375022	−0.3689713	0.35998444	−0.0368666	0.32907219	0.14877358	0.143483247	0.3344067
	9	−0.2877027	−0.0812976	0.58513537	0.11454134	0.04186864	−0.3814356	0.28795352	−0.00041389	−0.13335931
	10	−0.3073994	0.0111124	−0.0068885	0.04196657	−0.127565	0.14629639	−0.4081982	0.0030286	0.33443128
	11	0.06421859	−0.0919722	0.34881249	−0.0684058	−0.1412179	0.497947	0.19874151	0.071110753	0.298703303



(a) (b) (c)

Figure 15: Comparison between solutions obtained by the LeNN-WOA-NM algorithm under the influence of variations in β_2 on population densities of the prey-predator model. (a) Approximate solutions for population density of local prey. (b) Approximate solutions for population density of immigrant prey. (c) Approximate solutions for population density of the predator.

Table 33: Comparison between solutions obtained by the LeNN-WOA-NM algorithm and ode45 for the prey-predator model with $\gamma_1 = 0.015$.

t	$S(t)$	*ode*45	$X(t)$	*ode*45	$Y(t)$	*ode*45
0.0	0.50000029	0.5	0.49999971	0.5	0.50000002	0.5
0.2	0.52117967	0.52117476	0.4667562	0.46676439	0.59043325	0.59042518
0.4	0.54230262	0.54229891	0.42797497	0.42797925	0.69572032	0.695704
0.6	0.5631532	0.56315108	0.38434774	0.38435442	0.81726419	0.81725811
0.8	0.58351051	0.58350611	0.33705551	0.33706036	0.95637538	0.95637013
1.0	0.60315238	0.6031513	0.28769246	0.28769927	1.11420939	1.11419232

Table 34: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population densities of local prey, immigrant prey, and predator under the influence of variations in γ_1.

t	$S(t)$				$X(t)$				$Y(t)$			
	$\gamma_1 = 0.005$	$\gamma_1 = 0.015$	$\gamma_1 = 0.020$	$\gamma_1 = 0.025$	$\gamma_1 = 0.005$	$\gamma_1 = 0.015$	$\gamma_1 = 0.020$	$\gamma_1 = 0.025$	$\gamma_1 = 0.005$	$\gamma_1 = 0.015$	$\gamma_1 = 0.020$	$\gamma_1 = 0.025$
0.0	0.49999982	0.50000029	0.50000000	0.49999997	0.49999982	0.49999971	0.49999998	0.50000017	0.500000504	0.500000018	0.499999997	0.500000131
0.2	0.52174700	0.52117967	0.52089269	0.52060661	0.46675174	0.46675620	0.46676007	0.46676615	0.590449582	0.590433254	0.590411317	0.590394050
0.4	0.54359472	0.54230262	0.54165974	0.54101631	0.42795466	0.42797497	0.42797618	0.42799714	0.695846635	0.695720322	0.695630674	0.695558797
0.6	0.56533929	0.56315320	0.56206177	0.56097559	0.38430829	0.38434774	0.38437834	0.38439748	0.817678656	0.817264191	0.817046212	0.816837342
0.8	0.58680848	0.58351051	0.58186269	0.58022442	0.33695644	0.33705551	0.33711236	0.33716085	0.957316487	0.956375378	0.955894440	0.955419606
1.0	0.60782251	0.60315238	0.60083619	0.59852684	0.28751777	0.28769246	0.28777966	0.28787420	1.116062094	1.114209392	1.113259493	1.112332477

Table 35: Absolute errors obtained by the LeNN-WOA-NM algorithm for population densities of local prey, immigrant prey, and predator under the influence of variations in γ_1.

t	$S(t)$				$X(t)$				$Y(t)$			
	$\gamma_1 = 0.005$	$\gamma_1 = 0.015$	$\gamma_1 = 0.020$	$\gamma_1 = 0.025$	$\gamma_1 = 0.005$	$\gamma_1 = 0.015$	$\gamma_1 = 0.020$	$\gamma_1 = 0.025$	$\gamma_1 = 0.005$	$\gamma_1 = 0.015$	$\gamma_1 = 0.020$	$\gamma_1 = 0.025$
0.0	2.74E−09	8.47E−10	3.60E−10	7.80E−11	5.74E−09	2.00E−09	4.47E−09	4.19E−12	8.54E−10	6.23E−09	9.79E−12	5.10E−13
0.2	2.76E−09	4.69E−10	2.92E−10	1.04E−11	3.59E−09	8.16E−11	4.78E−09	5.77E−10	5.53E−10	6.28E−09	1.38E−11	8.72E−11
0.4	5.70E−10	4.68E−10	4.62E−13	3.63E−10	2.51E−09	3.02E−10	4.26E−10	2.89E−10	5.49E−10	8.44E−10	1.36E−12	2.06E−10
0.6	1.89E−09	1.24E−10	1.54E−10	2.49E−12	4.04E−11	3.74E−10	4.26E−09	2.55E−11	1.47E−12	3.54E−09	9.17E−11	1.14E−09
0.8	2.36E−09	1.80E−11	2.01E−12	8.31E−10	1.28E−09	4.93E−10	3.67E−09	5.03E−10	3.59E−10	3.49E−09	9.80E−11	4.43E−10
1.0	2.82E−09	1.85E−10	4.54E−12	1.17E−09	2.58E−10	1.22E−15	4.68E−09	1.08E−10	2.82E−10	4.48E−09	3.19E−10	1.19E−09

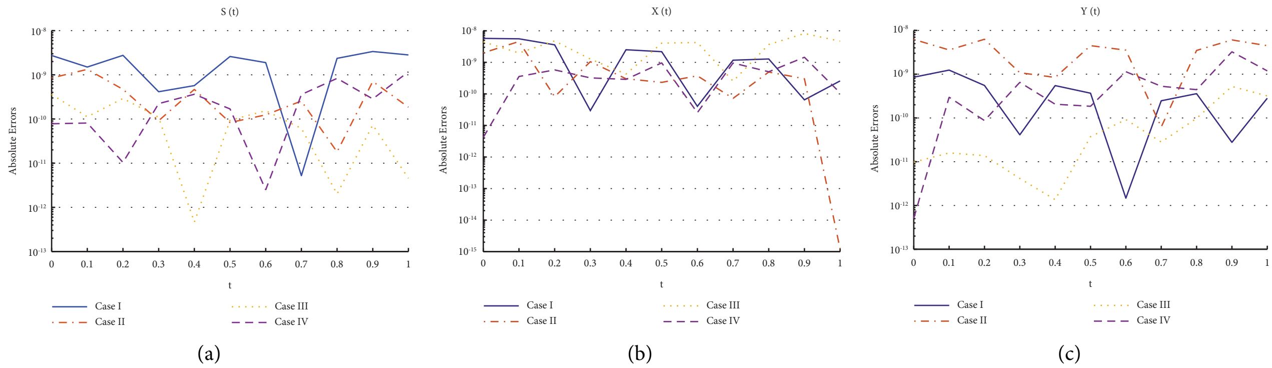


Figure 16: Absolute errors obtained by the LeNN-WOA-NM algorithm under the influence of variations in γ_1 on population densities of the prey-predator model. (a) Population density of local prey. (b) Population density of immigrant prey. (c) Population density of the predator.

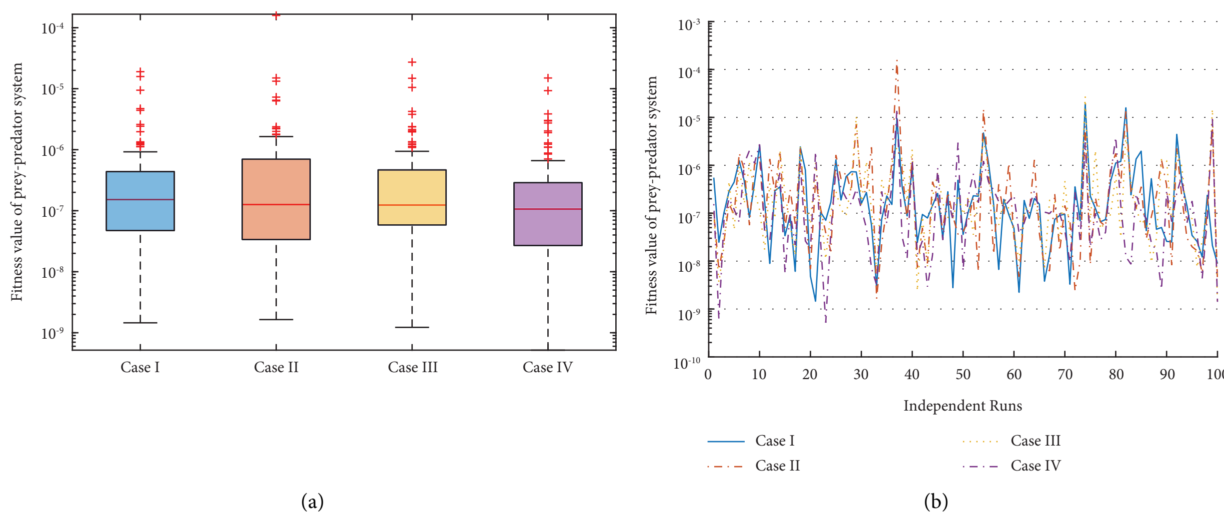


Figure 17: Comparison between the box plots and the convergence graph of fitness evaluation over 100 independent runs for the prey-predator model with variations in the catching rate of local prey. (a) Box plots for fitness evaluation of the prey-predator model with variations in γ_1. (b) Convergence of the fitness value of the prey-predator model with variations in γ_1.

Case II: $\gamma_2 = 0.8$

Case III: $\gamma_2 = 1.0$

Case IV: $\gamma_2 = 1.1$

The LeNN-WOA-NM algorithm is applied to prey-predator model equation (39) to study the influence of variations in the catching rate of immigrant prey. The results calculated by the designed algorithm are compared with those of the Runge–Kutta method using ode45 in MATLAB as shown in Table 39. Approximate solutions for population densities of local prey, immigrant prey, and the predator are given in Table 40. Absolute errors are presented in Table 41 and graphically shown in Figure 19. The minimum fitness values in Figure 20 reflect the accuracy of the solutions by the proposed algorithm. The convergence of our numerical approach is assessed by fitness values and statistical analysis of the performance indicators (MAD, TIC, and ENSE), see Table 42. Table 43 shows the

Table 36: Statistics of global performance indices for variations in the catching rate of local prey on the prey-predator model.

Cases	$S(t)$				$X(t)$				$Y(t)$			
	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE
I	2.50$E-06$	5.21$E-05$	2.69$E-05$	2.50$E-06$	9.20$E-07$	4.63$E-05$	3.33$E-05$	1.51$E-05$	9.19$E-07$	8.88$E-05$	3.10$E-05$	8.98$E-06$
II	7.32$E-06$	6.20$E-05$	3.14$E-05$	5.63$E-05$	2.43$E-06$	6.86$E-05$	4.92$E-05$	7.58$E-05$	1.44$E-06$	1.06$E-04$	3.80$E-05$	1.70$E-05$
III	2.86$E-06$	4.87$E-05$	2.48$E-05$	9.34$E-05$	7.60$E-07$	5.39$E-05$	3.89$E-05$	1.86$E-05$	1.39$E-06$	8.53$E-05$	3.09$E-05$	5.88$E-06$
IV	1.78$E-06$	4.62$E-05$	2.38$E-05$	7.52$E-05$	5.18$E-07$	4.32$E-05$	3.10$E-05$	1.46$E-05$	6.42$E-07$	8.01$E-05$	2.83$E-05$	9.08$E-06$

Table 37: Convergence analysis of population density of local prey, immigrant prey, and predator under the influence of variations in γ_1.

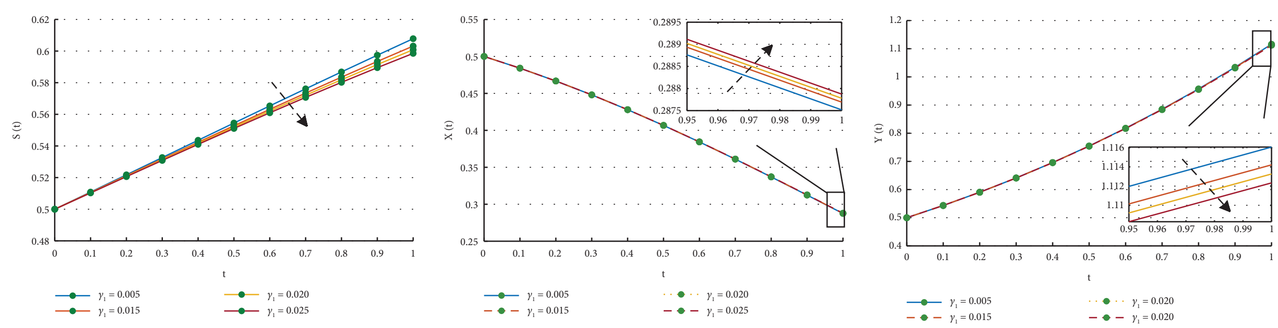
	Cases	FIT				MAD			TIC			ENSE		
		$\leq 10^{-6}$	$\leq 10^{-7}$	$\leq 10^{-8}$	$\leq 10^{-9}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-7}$
$S(t)$	I	95	70	20	5	100	85	21	100	94	37	85	56	21
	II	94	67	25	2	99	81	23	100	94	43	83	57	21
	III	95	71	17	2	100	90	19	100	96	42	91	55	19
	IV	97	78	29	6	100	91	26	100	97	42	90	57	26
$X(t)$	I	97	90	59	22	100	91	15	100	94	27	97	8	38
	II	97	86	56	24	99	84	16	99	91	28	96	78	40
	III	99	85	64	17	100	88	15	100	89	21	94	81	33
	IV	99	93	57	22	100	93	20	100	96	31	98	81	46
$Y(t)$	I	99	87	51	15	99	77	7	100	95	25	99	91	59
	II	96	78	48	15	99	73	8	100	93	30	98	87	62
	III	98	82	49	15	100	75	10	100	97	30	99	88	61
	IV	98	90	60	17	100	93	37	100	93	37	99	93	64

Table 38: Unknown parameters obtained by the LeNN-WOA-NM algorithm for the prey-predator model under the influence of variations in γ_1.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
Case I	1	0.10433411	−0.9093816	−0.0551384	0.10814164	−0.0975817	−0.4024759	0.11057901	0.030277215	−0.07683093
	2	−0.1015821	−0.2339063	0.11183609	−0.0079649	0.12409457	−0.0066366	0.03120447	0.065984151	0.169815479
	3	−0.1732219	0.05666544	−0.3418627	−0.1576219	0.20511634	0.23158339	−0.3370419	−0.15418238	0.194154313
	4	0.2870666	0.04587599	−0.075017	0.0317597	−0.1567915	−0.2960607	0.12661418	0.176984003	−0.08307132
	5	0.18918051	−0.2634862	0.09296959	0.17712714	0.06690293	0.05596124	0.06180474	−0.14442377	−0.98944725
	6	0.08011991	−0.0241843	0.04384194	0.09323427	−0.1029541	0.30404503	−0.1508149	−0.05051503	0.191092976
	7	−0.4734878	−0.1888967	0.13971948	0.24925106	0.00225215	0.00501846	−0.9616715	−0.19702187	0.011340631
	8	0.19753354	−0.0151743	−0.0358307	−0.2646482	−0.4318811	−0.2034545	−0.0870373	−0.08872014	0.079363347
	9	0.07352321	0.04374259	0.17641082	−0.004594	−0.0709633	−0.188571	0.01228411	0.051586673	−0.1707669
	10	0.05966701	−0.009466	0.11306597	−1.0101745	−0.1195556	0.30440396	−0.16551	0.072728886	0.05362364
	11	0.24419278	0.21523627	0.12501506	0.10887851	0.12112026	−0.1899001	−0.157371	−0.15906121	0.128530508
Case II	1	0.42562436	0.22405135	0.02503956	0.35262391	−0.2253664	0.26259455	0.27111686	0.328796691	0.228927872
	2	0.41666988	0.5078151	−0.1889093	0.12171391	0.20998105	0.3804858	0.30390747	0.389116532	0.460701796
	3	0.10577695	0.06002846	0.21768786	−0.5393398	0.39027559	0.13902226	0.47350419	0.465419864	0.661970007
	4	0.46517407	0.09698879	−0.1633478	0.11014297	0.19739907	0.6654931	0.05604469	−0.00392697	0.278777394
	5	0.60100029	0.01932057	0.56830498	0.2354314	0.41404057	0.20437334	−0.1267241	−0.13128894	0.293844939
	6	0.08127736	0.10628146	0.38610942	0.23539962	0.17187529	0.38579085	0.35225137	0.067649428	0.188489281
	7	0.35498411	0.04490212	0.04900664	0.06476587	0.35238074	0.50235806	0.36253639	0.203702979	−0.0792048
	8	0.38497077	0.01823575	0.34815044	−0.0755594	0.49478279	0.38035495	0.16102938	−0.12482814	0.340540011
	9	−0.0561765	0.73291315	−0.1890215	0.0053794	−0.4438896	0.46641096	0.2889986	−0.01721494	0.103497157
	10	$-8.42E-05$	−0.0191478	0.33057513	0.01138671	0.21012745	0.28013928	0.41790512	0.011022434	0.111373073
	11	0.03758884	−0.0090962	0.29841949	0.06181435	0.28211233	0.1044331	0.76029441	−0.05305741	0.400559476
Case III	1	0.34210927	−0.1686215	−0.0637152	0.23427117	0.25285975	0.39214738	0.26900094	0.565520288	0.485240258
	2	−0.0413096	−0.0081301	0.10614537	0.60106519	−0.0765896	0.42188566	0.34146251	0.575583706	0.081232416
	3	0.29904262	−0.075813	−0.1967563	0.45939119	0.15935586	−0.0737513	−0.0371428	0.395780195	0.526025075
	4	0.31966978	−0.0296846	0.13492819	0.74449918	0.13823099	0.14733057	0.31296573	0.170356341	0.449030416
	5	0.38511171	0.02742915	0.19245868	0.17480946	0.26067733	0.19256129	−0.1966744	0.026652715	0.648556737
	6	0.15405406	−0.085595	0.03900298	0.20316026	−0.0939363	0.36999635	0.14116303	0.18349354	0.157284014
	7	0.01160236	0.19042424	−0.0099165	−0.1955586	−0.0145249	0.5115481	0.09234283	0.050183053	0.529576694
	8	−0.0119424	−0.0017222	0.19894821	−0.0459019	0.16995456	0.09912122	0.44277162	0.268197932	0.360452198
	9	−0.013509	0.02826738	0.13366696	−0.0986121	0.12555621	0.29387726	−0.0351511	0.15868406	−0.02656018
	10	0.36233153	−0.0504481	0.46422188	0.33154131	0.04488349	0.44423777	0.18233457	0.002699819	0.25017744
	11	0.20780852	0.06187582	0.19012834	0.47309423	−0.0001455	0.28544612	0.32655747	0.232204493	0.038376682

Table 38: Continued.

	Index	$S(t)$ ζ_n	$S(t)$ ψ_n	$S(t)$ θ_n	$X(t)$ ζ_n	$X(t)$ ψ_n	$X(t)$ θ_n	$Y(t)$ ζ_n	$Y(t)$ ψ_n	$Y(t)$ θ_n
Case IV	1	0.08743197	−0.0274586	0.13838454	0.10216597	−0.6205394	−1.1625473	0.08890874	−0.46821489	0.031906641
	2	0.85138918	−0.2596089	0.27697631	−0.0395679	−0.3190484	0.47807164	−1.0766899	0.194522468	0.051473955
	3	0.32484395	−0.0320128	0.04910259	0.59721285	−0.7402654	−0.5159275	−0.5457082	0.584514565	0.334208342
	4	1.26539529	0.0637656	−0.3298542	0.00746795	0.12675579	−0.2119938	0.0689466	0.078060063	0.341243622
	5	0.27124475	−0.0145706	−0.8795409	−0.2258574	−1.0489144	0.18138843	−0.7172783	−0.44654588	0.288352353
	6	−0.0615221	−0.0357181	0.05167451	−0.0093528	−0.9076303	0.00529929	0.11164685	0.084161905	0.021304977
	7	−0.9330419	0.00587016	0.08238699	−0.0038462	0.5318456	−0.0029243	−0.5624983	0.07126723	−0.07194593
	8	−0.2222851	0.15964489	0.15929347	0.02677628	−0.0859581	0.71257834	−0.1564259	0.011627909	−0.01141564
	9	−0.0746732	0.03595603	−0.1688295	0.91853869	0.0432707	−0.1996406	0.9708232	−0.20718687	0.491048316
	10	0.11105985	0.03339826	−0.0412419	−0.9111904	−0.0757299	0.38318296	0.14046109	0.068233399	0.447903219
	11	−0.9956577	−0.0325751	0.95862042	0.01858614	−0.0328286	0.2342197	−0.8047726	−0.03547963	0.018056695



(a) (b) (c)

Figure 18: Comparison between solutions obtained by the LeNN-WOA-NM algorithm under the influence of variations in γ_1 on population densities of the prey-predator model. (a) Approximate solutions for population density of local prey. (b) Approximate solutions for population density of immigrant prey. (c) Approximate solutions for population density of the predator.

Table 39: Comparison between solutions obtained by the LeNN-WOA-NM algorithm and ode45 for the prey-predator model with $\gamma_2 = 0.8$.

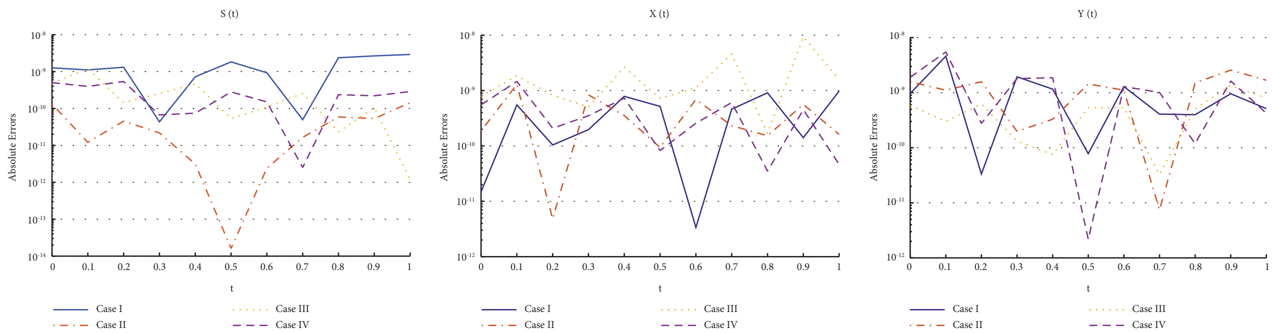
t	$S(t)$	$ode45$	$X(t)$	$ode45$	$Y(t)$	$ode45$
0.0	0.50000006	0.5	0.5000005	0.5	0.49999982	0.5
0.2	0.52151097	0.52117476	0.47186632	0.46676439	0.59068568	0.59042518
0.4	0.54316182	0.54229891	0.438185	0.42797925	0.69694028	0.695704
0.6	0.56475603	0.56315108	0.39935526	0.38435442	0.82053699	0.81725811
0.8	0.58608332	0.58350611	0.35617116	0.33706036	0.96319165	0.95637013
1.0	0.60693367	0.6031513	0.30986509	0.28769927	1.12649634	1.11419232

Table 40: Approximate solutions obtained by the LeNN-WOA-NM algorithm for population densities of local prey, immigrant prey, and predator under the influence of γ_2.

t	$S(t)$				$X(t)$				$Y(t)$			
	$\gamma_2 = 0.7$	$\gamma_2 = 0.8$	$\gamma_2 = 1.0$	$\gamma_2 = 1.1$	$\gamma_2 = 0.7$	$\gamma_2 = 0.8$	$\gamma_2 = 1.0$	$\gamma_2 = 1.1$	$\gamma_2 = 0.7$	$\gamma_2 = 0.8$	$\gamma_2 = 1.0$	$\gamma_2 = 1.1$
0.0	0.50000038	0.50000006	0.49999999	0.49999984	0.50000048	0.5000005	0.50000004	0.49999999	0.499999779	0.499999818	0.499999984	0.499999697
0.2	0.5215693	0.52151097	0.52141035	0.52135099	0.47702117	0.47186632	0.46172579	0.45673788	0.590933443	0.590685682	0.590202752	0.589971805
0.4	0.5433938	0.54316182	0.542727	0.54250628	0.44865641	0.438185	0.418006	0.40828236	0.69811051	0.696940282	0.694649876	0.693532481
0.6	0.56528285	0.56475603	0.56374521	0.56325551	0.41500305	0.39935526	0.3699049	0.35605878	0.823696938	0.820536986	0.814493466	0.811599233
0.8	0.58704862	0.58608332	0.584264	0.58339627	0.37651173	0.35617116	0.31896825	0.30195215	0.969820246	0.963191645	0.950766655	0.944938366
1.0	0.60847136	0.60693367	0.60409654	0.60277219	0.3340243	0.30986509	0.26708179	0.24814624	1.138532264	1.126496337	1.104411161	1.094260679

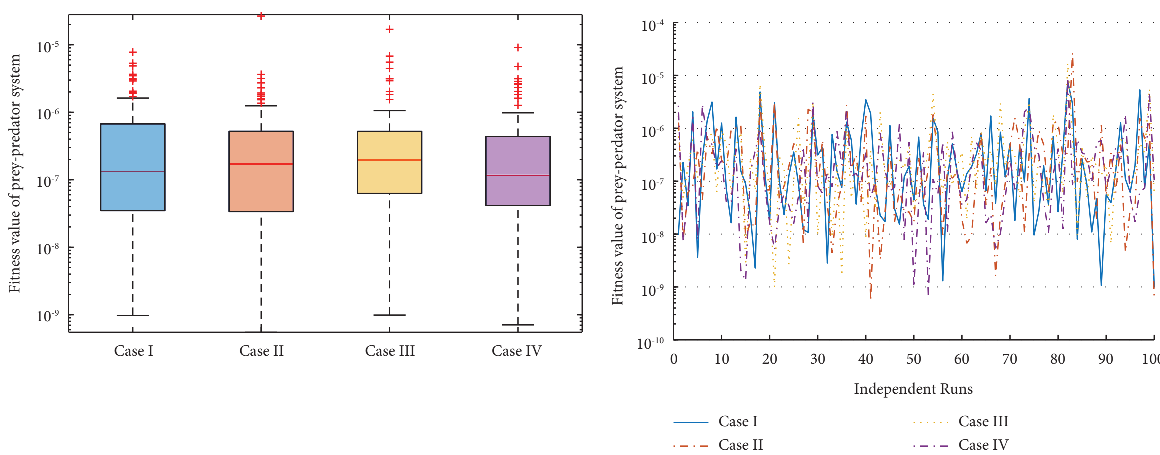
TABLE 41: Absolute errors obtained by the LeNN-WOA-NM algorithm for population densities of local prey, immigrant prey, and predator under the influence of γ_2.

t	$S(t)$				$X(t)$				$Y(t)$			
	$\gamma_2 = 0.7$	$\gamma_2 = 0.8$	$\gamma_2 = 1.0$	$\gamma_2 = 1.1$	$\gamma_2 = 0.7$	$\gamma_2 = 0.8$	$\gamma_2 = 1.0$	$\gamma_2 = 1.1$	$\gamma_2 = 0.7$	$\gamma_2 = 0.8$	$\gamma_2 = 1.0$	$\gamma_2 = 1.1$
0.0	1.26E − 09	1.32E − 10	4.79E − 10	5.02E − 10	1.48E − 11	1.87E − 10	7.79E − 10	5.53E − 10	9.67E − 10	1.58E − 09	5.89E − 10	1.90E − 09
0.2	1.31E − 09	4.54E − 11	1.43E − 10	5.38E − 10	1.04E − 10	4.83E − 12	8.26E − 10	2.09E − 10	3.34E − 11	1.58E − 09	6.20E − 10	2.77E − 10
0.4	7.22E − 10	3.13E − 12	4.65E − 10	7.47E − 11	7.86E − 10	3.53E − 10	2.65E − 09	7.30E − 10	1.16E − 09	3.29E − 10	7.51E − 11	1.87E − 09
0.6	9.27E − 10	2.41E − 12	1.09E − 10	1.51E − 10	3.37E − 12	6.92E − 10	1.12E − 09	2.57E − 10	1.29E − 09	1.11E − 09	5.35E − 10	1.29E − 09
0.8	2.37E − 09	5.90E − 11	2.34E − 11	2.37E − 10	9.11E − 10	1.53E − 10	1.69E − 10	3.50E − 11	3.98E − 10	1.50E − 09	5.10E − 10	1.19E − 10
1.0	2.91E − 09	1.44E − 10	1.20E − 12	2.88E − 10	9.82E − 10	1.61E − 10	1.58E − 09	4.63E − 11	5.13E − 10	1.68E − 09	7.98E − 10	4.24E − 10



(a) (b) (c)

Figure 19: Absolute errors obtained by the LeNN-WOA-NM algorithm under the influence of variations in γ_2 on population densities of the prey-predator model. (a) Population density of local prey. (b) Population density of immigrant prey. (c) Population density of the predator.



(a) (b)

Figure 20: Comparison between the box plots and convergence graphs of fitness evaluation over 100 independent runs for the prey-predator model with variations in the catching rate of immigrant prey. (a) Box plots for fitness evaluation of the prey-predator model with variations in γ_2. (b) Convergence of the fitness value of the prey-predator model with variations in γ_2.

Table 42: Convergence analysis of population density of local prey, immigrant prey, and predator under the influence of variations in γ_2.

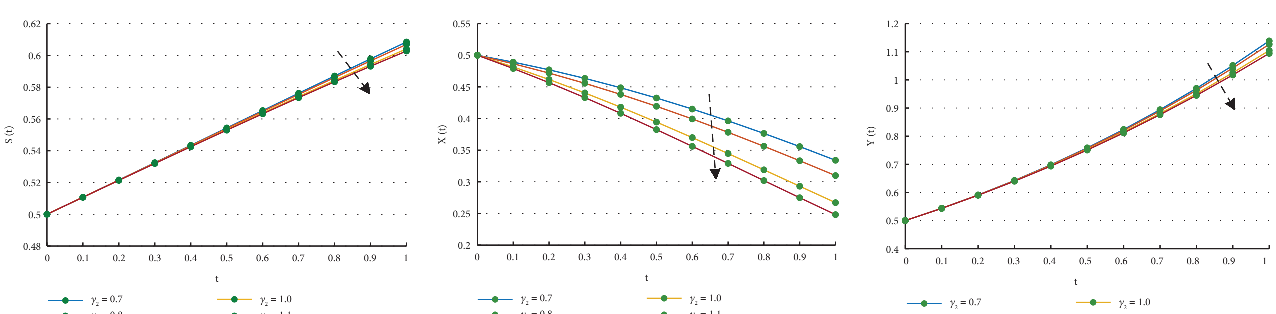
	Cases	FIT				MAD			TIC			ENSE		
		$\le 10^{-6}$	$\le 10^{-7}$	$\le 10^{-8}$	$\le 10^{-9}$	$\le 10^{-4}$	$\le 10^{-5}$	$\le 10^{-6}$	$\le 10^{-4}$	$\le 10^{-5}$	$\le 10^{-6}$	$\le 10^{-5}$	$\le 10^{-6}$	$\le 10^{-7}$
$S(t)$	I	95	65	24	5	100	88	20	100	95	41	89	59	20
	II	98	64	24	4	100	92	18	100	99	45	93	60	18
	III	86	63	12	4	100	90	16	100	95	32	90	55	16
	IV	98	67	20	4	100	92	21	100	97	44	92	62	21
$X(t)$	I	99	85	52	22	100	85	20	100	92	30	94	69	32
	II	100	87	57	22	100	87	23	100	91	27	98	73	33
	III	98	83	46	13	100	82	10	100	93	16	98	71	28
	IV	99	90	54	19	100	94	17	100	97	23	99	82	44
$Y(t)$	I	98	84	44	17	100	77	11	100	92	28	100	87	57
	II	99	81	47	18	99	75	13	100	97	31	99	87	55
	III	99	84	41	15	100	75	6	100	94	26	99	89	56
	IV	99	85	50	14	100	72	9	100	94	34	100	87	55

Table 43: Statistics of global performance indices for variations in the catching rate of local prey on the prey-predator model.

Cases	$S(t)$				$X(t)$				$Y(t)$			
	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE	GFIT	GMAD	GTIC	GENSE
I	$1.98E-06$	$4.80E-05$	$2.43E-05$	$5.78E-05$	$5.99E-07$	$5.04E-05$	$3.39E-05$	$2.68E-05$	$9.28E-07$	$9.13E-05$	$3.24E-05$	$5.44E-06$
II	$2.17E-06$	$3.83E-05$	$1.95E-05$	$2.91E-05$	$5.21E-07$	$4.70E-05$	$3.30E-05$	$1.44E-05$	$1.36E-06$	$9.17E-05$	$3.25E-05$	$1.01E-05$
III	$2.16E-06$	$4.95E-05$	$2.52E-05$	$5.55E-05$	$1.01E-06$	$5.58E-05$	$4.16E-05$	$1.53E-05$	$7.10E-07$	$8.79E-05$	$3.17E-05$	$6.81E-06$
IV	$1.55E-06$	$4.01E-05$	$2.00E-05$	$4.85E-05$	$5.75E-07$	$4.49E-05$	$3.40E-05$	$7.86E-06$	$7.28E-07$	$8.59E-05$	$3.08E-05$	$5.50E-06$

Table 44: Unknown parameters obtained by the LeNN-WOA-NM algorithm for the prey-predator model under the influence of variations in γ_1.

	Index	$S(t)$			$X(t)$			$Y(t)$		
		ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
	1	0.27328231	−0.0407082	−0.0036568	0.19095405	0.19619913	0.25334058	0.15448032	0.339199456	0.646260329
	2	0.00203693	0.00761269	0.24655049	0.67156041	−0.1202185	0.41029823	0.35456658	0.167129547	−0.02220108
	3	0.22854565	−0.1965182	−0.0799706	0.423423	0.03898304	−0.1160494	−0.1932211	0.327298457	0.589634109
	4	0.43068766	−0.0047135	−0.0067749	0.7614451	0.24436812	0.10386	0.45804558	−0.10504105	0.096632479
	5	0.42381358	−0.1502652	0.51827442	0.10757996	0.24696908	0.11302049	−0.0490828	0.054497193	0.541464355
Case I	6	0.03064345	0.387909	−0.0908338	0.20021796	−0.1093334	0.49803749	0.08502235	0.292312181	0.168951799
	7	−0.0256272	0.18134944	0.04903339	−0.1439181	0.01678089	0.01522287	0.12228645	0.040610745	0.504241595
	8	−0.0029234	0.05369165	0.16118629	0.09238371	0.35203171	−0.0742458	0.58296355	0.326394622	0.311232469
	9	0.01433551	−0.0062826	0.11461784	0.00070281	0.08170655	0.27442616	−0.0624975	0.046728012	−0.00253833
	10	0.52132655	−0.0745824	0.32373133	0.31625889	0.06469584	0.35518716	0.43265994	0.001675421	−0.07508196
	11	0.04700892	−0.1431138	0.19096034	0.61086683	0.05879562	0.07145247	0.31602732	0.294568546	0.005492958
	1	0.22888098	0.15662727	0.72118398	0.66396301	0.48400554	−0.0014987	0.2930431	−0.36246252	−0.19827556
	2	0.32039169	0.21349716	0.26632024	−0.1848203	−0.0673308	−0.0648246	0.13125299	0.041657028	0.029954374
	3	−0.0829388	0.03094832	0.00303522	0.62951247	−0.1925438	0.24736655	−0.1963121	0.431531435	0.708095767
	4	−0.1658907	−0.1447463	−0.6955688	−0.232498	−0.2616095	0.27025227	0.22782911	−0.28937041	−0.12994496
	5	−0.0174758	0.23215182	0.04290016	0.12334482	−0.2128439	−0.0360696	−0.1459274	0.143383282	0.127654212
Case II	6	−0.0712296	−0.3626918	−0.0348668	0.2600438	0.13168403	0.45131095	0.73463526	−0.09109894	0.100058657
	7	−0.0244551	−0.0112202	−0.1246274	−0.3445732	0.31169855	−0.0680798	0.14809738	0.029526212	0.176039899
	8	−0.2963948	0.27684365	0.02435043	−0.1570692	−0.1058024	0.01048308	−0.0287022	0.052204737	0.500290252
	9	−0.0250215	−0.1024261	0.05651485	0.61926963	−0.205578	0.43411127	−0.3748423	0.146360873	−0.41978655
	10	−0.2847311	0.13336075	−0.0903463	−0.1464593	−0.3554768	−0.0475072	0.16657978	−0.04184543	−0.04410262
	11	−0.0012914	0.02947063	−0.2027891	−0.0119255	0.47238101	0.09448735	0.42979806	−0.05005385	−0.00793862
	1	0.17136911	−0.6220119	−0.0659451	0.16001338	0.06604026	−0.6259462	0.41743442	0.34745146	0.123402726
	2	0.13248093	0.22407198	0.16846536	−0.0174004	0.46048206	0.16172196	0.6363567	0.027892663	0.349217115
	3	−0.029666	0.0085174	−0.4187633	−0.0573402	0.29673852	0.13113638	−0.7934564	−0.18170045	0.087257639
	4	0.50916399	0.04634821	−0.1437019	0.08923057	−0.2567021	−0.2934746	0.26333479	−0.21308607	−0.02505351
	5	0.28783823	−0.2662163	0.0766028	0.09914485	0.57665811	0.04300083	0.20662283	−0.48985711	−1.09740743
Case III	6	0.08917337	−0.1050009	0.23818256	0.02035607	−0.1072192	0.2659477	−0.0029764	−0.13148311	0.313665366
	7	−0.7503685	−0.1704471	0.18719634	−0.0233035	−0.0088938	0.26343414	−1.0737247	−0.19018699	0.280100524
	8	0.34230089	0.00558946	−0.1427919	−0.2957063	−0.6490481	0.10546459	0.27322547	0.041606539	−0.10278157
	9	0.25879736	0.21953927	0.11052919	0.00206003	−0.0680416	−0.2189914	0.0378666	0.055450179	−0.26693807
	10	0.11482869	0.04256621	0.24740165	−1.0210288	0.10826852	0.1081171	0.28972162	0.107963625	−0.01416421
	11	0.21645891	0.26405307	−0.2682646	−0.0251217	0.2673504	−0.2407415	−0.3201356	−0.12644907	0.363928738
	1	0.16321044	1.00101739	0.44610851	0.19117524	−0.1869131	0.03284449	0.48709406	0.492839005	0.068811431
	2	0.05034014	0.27448502	0.05595427	0.34759862	−0.0428559	−0.4656632	0.41211291	0.550350162	0.261254457
	3	0.02309873	0.00700133	−0.3300406	−0.032857	−0.2059139	−0.1572132	0.43069278	0.119708938	0.13204548
	4	0.60643185	−0.1677085	0.01351505	0.01269345	0.07414495	−0.2533687	−0.1990881	0.074189084	−0.00874219
	5	0.06424168	−0.0394268	0.04062341	0.10346071	0.24294981	−0.1406991	0.1530591	0.00796601	0.142324569
Case IV	6	−0.1607398	−0.0067892	−0.1142057	−0.3542569	−0.317926	0.89646875	0.19029585	−0.0777253	0.138215439
	7	−0.0249375	−0.142179	0.21181449	0.00745531	−0.0807929	−0.0606121	−0.3138917	−0.04179489	−0.08395995
	8	0.01271854	0.07904033	−0.2032421	0.30678256	−0.0970233	−0.0973761	−0.2681465	−0.26023333	−0.05748785
	9	0.15729166	0.02004037	0.08441775	0.50899351	−0.0107112	0.1654774	0.05321702	−0.55038421	−0.01142195
	10	0.45526814	0.30848224	0.03885589	−0.067078	0.03709569	−0.1103419	0.00132508	−0.0601008	−0.00809993
	11	0.00144682	−0.0160469	−0.1485655	0.07678165	−0.0117808	−0.039538	−0.1553724	0.090185082	−0.05537005



(a) (b) (c)

Figure 21: Comparison between solutions obtained by the LeNN-WOA-NM algorithm under the influence of variations in γ_2 on population densities of the prey-predator model. (a) Approximate solutions for population density of local prey. (b) Approximate solutions for population density of immigrant prey. (c) Approximate solutions for population density of the predator.

correctness of the proposed algorithm to tackle real-world problems. Trained neurons in LeNN are shown in Table 44. From Figure 21, the following conclusions can be drawn:

(i) Population densities of local prey, immigrant prey, and the predator vary inversely with variation in the catching rate of immigrant prey γ_2.

8. Conclusion

In this paper, we have analyzed the system of the singular differential equation (SDE) representing the phenomena of the prey-predator model with immigrant prey. Generally, solving SDE is one of the challenging tasks, and therefore, we have designed a novel soft computing technique known as a LeNN-WOA-NM algorithm. Weighted Legendre polynomials are used to model approximate series solutions for the prey-predator model with variations in various coefficients, including the growth rate of local and immigrant prey (α_1, α_2), force interaction between local and immigrant prey (β_1, β_2), and the catching rate of local and immigrant prey γ_1, γ_2. We summarize our findings as follows:

(i) Variations in $\alpha_1, \alpha_2, \beta_1$, and β_2 has a direct impact on population density of local prey $S(t)$, while population density of local prey is inversely affected by variations in γ_1 and γ_2.

(ii) Variations in $\alpha_1, \alpha_2, \beta_2$, and γ_2 has an inverse impact on population density of immigrant prey $X(t)$, while population density of immigrant prey is directly affected by variations in β_1 and γ_2.

(iii) Variations in α_1, α_2 and β_1 has a direct impact on population density of the predator $S(t)$, while population density of the predator is inversely affected by variations in γ_1, β_2, and γ_2.

(iv) Approximate solutions obtained by the LeNN-WOA-NM algorithm are compared with those obtained by the homotopy perturbation method and MATLAB solver ode45. The results show the dominance of the proposed technique.

(v) Lower absolute errors in our solutions and convergence analysis of fitness evaluation, MAD, TIC, and ENSE show the accuracy and robustness of the proposed algorithm for obtaining solutions to real-world problems.

In future, this approach can be utilized to solve the complex nonlinear systems of fractional differential equations characterizing real-world problems, for instance, anomalous diffusion of contaminant from the fracture into the porous rock matrix, microbial survival and growth curves, smoking dynamics, parametric identification of Hammerstein systems with time delay, and asymmetric dead zones.

Abbreviations

LeNN: Legendre neural network
NM: Nelder–Mead
MAD: Mean absolute diviation
TIC: Theil's inequality coefficient
NSE: Nash–Sutcliffe efficiency
ENSE: Error in Nash–Sutcliffe efficiency
ANNs: Artificial neural networks
WOA: Whale optimization algorithm
α_1: Intrinsic growth rate of local prey
α_2: Increasing rate of immigrant prey
β_1, β_2: Force of interaction between local prey and immigrant prey
k_1: Carrying capacity of local prey
k_2: Carrying capacity of immigrant prey
c_1, c_2: Intrinsic growth rate of the predator population
μ_1, μ_2: Suffering loss of the predator population
γ_1: Catching rate of local prey
γ_2: Catching rate of immigrant prey.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this study.

References

[1] X. Liu and L. Chen, "Complex dynamics of Holling type II Lotka–Volterra predator–prey system with impulsive perturbations on the predator," *Chaos, Solitons & Fractals*, vol. 16, no. 2, pp. 311–320, 2003.
[2] C. S. Holling, "The functional response of invertebrate predators to prey density," *Memoirs of the Entomological Society of Canada*, vol. 98, no. S48, pp. 5–86, 1966.
[3] M. Danca, S. Codreanu, and B. Bako, "Detailed analysis of a nonlinear prey-predator model," *Journal of Biological Physics*, vol. 23, no. 1, pp. 11–20, 1997.
[4] F. J. Solis, "Self-limitation in a discrete predator–prey model," *Mathematical and Computer Modelling*, vol. 48, no. 1-2, pp. 191–196, 2008.
[5] M. Fan and K. Wang, "Periodicity in a delayed ratio-dependent predator–prey system," *Journal of Mathematical Analysis and Applications*, vol. 262, no. 1, pp. 179–190, 2001.
[6] H. Freedman, *Deterministic Mathematical Models in Population Ecology*, Marcal Dekker, New York, USA, 1980.
[7] H. Freedman and R. Mathsen, "Persistence in predator-prey systems with ratio-dependent predator influence," *Bulletin of Mathematical Biology*, vol. 55, no. 4, pp. 817–827, 1993.
[8] Y. Cai, M. Banerjee, Y. Kang, and W. Wang, "Spatiotemporal complexity in a predator–prey model with weak allee effects," *Mathematical Biosciences and Engineering*, vol. 11, no. 6, pp. 1247–1274, 2014.
[9] M. Bandyopadhyay and S. Banerjee, "A stage-structured prey–predator model with discrete time delay," *Applied Mathematics and Computation*, vol. 182, no. 2, pp. 1385–1398, 2006.
[10] Z. Jing and J. Yang, "Bifurcation and chaos in discrete-time predator–prey system," *Chaos, Solitons & Fractals*, vol. 27, no. 1, pp. 259–277, 2006.

[11] X. Liu and D. Xiao, "Complex dynamic behaviors of a discrete-time predator–prey system," *Chaos, Solitons & Fractals*, vol. 32, no. 1, pp. 80–94, 2007.
[12] Y. Huang, F. Chen, and L. Zhong, "Stability analysis of a prey–predator model with Holling type III response function incorporating a prey refuge," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 672–683, 2006.
[13] Q. Din, "Complexity and chaos control in a discrete-time prey-predator model," *Communications in Nonlinear Science and Numerical Simulation*, vol. 49, pp. 113–134, 2017.
[14] V. Weide, M. C. Varriale, and F. M. Hilker, "Hydra effect and paradox of enrichment in discrete-time predator-prey models," *Mathematical Biosciences*, vol. 310, pp. 120–127, 2019.
[15] Z. Gong, R. Hamazaki, and M. Ueda, "Discrete time-crystalline order in cavity and circuit QED systems," *Physical Review Letters*, vol. 120, no. 4, Article ID 040404, 2018.
[16] T. Huang, H. Zhang, H. Yang, N. Wang, and F. Zhang, "Complex patterns in a space-and time-discrete predator-prey model with Beddington-DeAngelis functional response," *Communications in Nonlinear Science and Numerical Simulation*, vol. 43, pp. 182–199, 2017.
[17] K. P. Hadeler and H. I. Freedman, "Predator-prey populations with parasitic infection," *Journal of Mathematical Biology*, vol. 27, no. 6, pp. 609–631, 1989.
[18] B. Ghanbari and D. Kumar, "Numerical solution of predator-prey model with Beddington-DeAngelis functional response and fractional derivatives with Mittag-Leffler kernel," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 6, Article ID 063103, 2019.
[19] M. Perc and A. Szolnoki, "Noise-guided evolution within cyclical interactions," *New Journal of Physics*, vol. 9, no. 8, p. 267, 2007.
[20] G. Q. Sun, Z. Jin, L. Li, and Q. X. Liu, "The role of noise in a predator–prey model with Allee effect," *Journal of Biological Physics*, vol. 35, no. 2, pp. 185–196, 2009.
[21] J. Hofbauer, K. Sigmund, and others, *Evolutionary Games and Population Dynamics*, Cambridge University Press, USA, 1998.
[22] A. Kilbas and J. Trujillo, "Differential equations of fractional order: methods, results and problems. II," *Applicable Analysis*, vol. 81, no. 2, pp. 435–493, 2002.
[23] J. H. He, "A tutorial review on fractal spacetime and fractional calculus," *International Journal of Theoretical Physics*, vol. 53, no. 11, pp. 3698–3718, 2014.
[24] R. Gorenflo, A. A. Kilbas, F. Mainardi, S. V. Rogosin, and others, *Mittag-Leffler Functions, Related Topics and Applications*, Springer, USA, 2020.
[25] S. Paul, S. P. Mondal, and P. Bhattacharya, "Numerical solution of Lotka Volterra prey predator model by using Runge–Kutta–Fehlberg method and Laplace Adomian decomposition method," *Alexandria Engineering Journal*, vol. 55, no. 1, pp. 613–617, 2016.
[26] S. S. Ray, "A new coupled fractional reduced differential transform method for the numerical solution of fractional predator-prey system," *Computer Modeling in Engineering and Sciences*, vol. 105, pp. 231–249, 2015.
[27] Z. Anastassi and T. Simos, "An optimized Runge–Kutta method for the solution of orbital problems," *Journal of Computational and Applied Mathematics*, vol. 175, pp. 1–9, 2005.
[28] M. R. Garvie, J. Burkardt, and J. Morgan, "Simple finite element methods for approximating predator–prey dynamics in two dimensions using Matlab," *Bulletin of Mathematical Biology*, vol. 77, no. 3, pp. 548–578, 2015.
[29] N. Bildik and S. Deniz, "The use of Sumudu decomposition method for solving predator-prey systems," *Math. Sci. Lett*, vol. 5, no. 3, pp. 285–289, 2016.
[30] I. Gad and P. Novati, "The solution of fractional order epidemic model by implicit Adams methods," *Applied Mathematical Modelling*, vol. 43, pp. 78–84, 2016.
[31] I. Bashkirtseva and L. Ryashko, "Analysis of the noise-induced regimes in Ricker population model with Allee effect via confidence domains technique," *BioMed Research International*, vol. 2014, Article ID 346239, 7 pages, 2014.
[32] J. S. Yu and J. J. Yu, "Homotopy analysis method for a prey-predator system with holling IV functional response," *Applied Mechanics and Materials*, vol. 687-691, pp. 1286–1291, 2014.
[33] O. Dababneh, T. Kipouros, and J. F. Whidborne, "Application of an efficient gradient-based optimization strategy for aircraft wing structures," *Aerospace*, vol. 5, no. 1, p. 3, 2018.
[34] N. A. Khan, M. Sulaiman, A. J. Aljohani, M. A. Bakar, and others, "Mathematical models of CBSC over wireless channels and their analysis by using the LeNN-WOA-NM algorithm," *Engineering Applications of Artificial Intelligence*, vol. 107, Article ID 104537, 2022.
[35] S. B. Alagarsamy and K. Murugan, "Multimodal of ear and face biometric recognition using adaptive approach Runge–Kutta threshold segmentation and classifier with score level fusion," *Wireless Personal Communications*, vol. 124, no. 2, pp. 1061–1080, 2022.
[36] N. A. Khan, M. Sulaiman, C. A. Tavera Romero, and F. K. Alarfaj, "Numerical analysis of electrohydrodynamic flow in a circular cylindrical conduit by using neuro evolutionary technique," *Energies*, vol. 14, no. 22, p. 7774, 2021.
[37] N. A. Khan, F. S. Alshammari, C. A. T. Romero, M. Sulaiman, and G. Laouini, "Mathematical analysis of reaction–diffusion equations modeling the michaelis–menten kinetics in a micro-disk biosensor," *Molecules*, vol. 26, no. 23, p. 7310, 2021.
[38] N. A. Khan, F. S. Alshammari, C. A. Tavera Romero, M. Sulaiman, and S. Mirjalili, "An optimistic solver for the mathematical model of the flow of Johnson segalman fluid on the surface of an infinitely long vertical cylinder," *Materials*, vol. 14, no. 24, p. 7798, 2021.
[39] Y. Cui, Y. Hong, N. A. Khan, and M. Sulaiman, "Application of soft computing paradigm to large deformation analysis of cantilever beam under point load," *Complexity*, vol. 2021, pp. 2021–19, 2021.
[40] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
[41] S. Singer and J. Nelder, "Nelder-Mead algorithm," *Scholarpedia*, vol. 4, no. 7, p. 2928, 2009.
[42] R. Jovanovic, S. Kais, and F. H. Alharbi, *Cuckoo Search Inspired Hybridization of the Nelder-mead Simplex Algorithm Applied to Optimization of Photovoltaic Cells*, arXiv preprint arXiv:1411.0217, 2014.
[43] N. A. Khan, M. Sulaiman, A. J. Aljohani, P. Kumam, and H. Alrabaiah, *Analysis of Multi-phase Flow through Porous media for Imbibition Phenomena by Using the LeNN-WOA-NM Algorithm*, IEEE Access, USA, 2020.
[44] K. Renganathan, V. Ananthaswamy, and S. Narmatha, "Mathematical analysis of prey predator system with immigrant prey using a new approach to Homotopy perturbation method," *Materials Today Proceedings*, vol. 37, pp. 1183–1189, 2021.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 9089528, 12 pages
https://doi.org/10.1155/2022/9089528

Research Article

The Application of Convolutional Neural Network Combined with Fuzzy Algorithm in Colorectal Endoscopy for Tumor Assessment

Te-Jen Su,[1,2] Feng-Chun Lee,[1,3] Cheuk-Kwan Sun,[4,5] Fu-Xiang Ke,[6] Shih-Ming Wang ,[7] and Ming-Chih Huang[8]

[1]*Department of Electronics, National Kaohsiung University of Science and Technology, Kaohsiung 807, Taiwan*
[2]*Graduate Institute of Clinical Medicine, Kaohsiung Medical University School of Dentistry, Kaohsiung 807, Taiwan*
[3]*Kaohsiung Change Gung Memorial Hospital, Kaohsiung 833, Taiwan*
[4]*Department of Emergency Medicine, E-Da Hospital, Kaohsiung, Taiwan*
[5]*School of Medicine for International Students, College of Medicine, I-Shou University, Kaohsiung, Taiwan*
[6]*Department of Electronics, National Kaohsiung University of Science and Technology, Kaohsiung 807, Taiwan*
[7]*Department of Computer Science and Information Engineering, Cheng Shiu University, Kaohsiung 833, Taiwan*
[8]*Department of Electronics, National Kaohsiung University of Science and Technology, Kaohsiung 807, Taiwan*

Correspondence should be addressed to Shih-Ming Wang; k1115@gcloud.csu.edu.tw

Received 21 January 2022; Revised 30 April 2022; Accepted 31 May 2022; Published 7 July 2022

Academic Editor: Chenguang Yang

According to the Global Cancer Statistics 2020 published in the official journal of the American Cancer Society (ACS), colorectal cancer ranked 4th in incidence and 2nd in mortality, and the 2018 Cancer Registry Report of Taiwan Health Promotion Administration showed that colorectal cancer ranked 2nd in incidence and 3rd in mortality. With the rapid evolution of the times, the lifestyles of the people have shifted from what they used to be. In addition to uncontrollable factors such as family genetic disorders, diet, and bad habits, life stress may lead to an unhealthy body mass index (BMI), which, together with aging, increases the incidence of colorectal cancer. In this study, the convolutional neural network was used to assess the risk of tumor in the colon by colonoscopy. The endoscopic images of the colon, which were classified into three categories of healthy (normal), benign tumor, and malignant tumor, were adopted as training data. When this method is combined with the patient's physical data, the risk cancer can be calculated by the fuzzy algorithm. Based on the result of this study, the accuracy of the tumor profile by colonoscopy, that is, 81.6%, is more precise than that of colorectal cancer tumor analysis studies in the recent literature. The proposed method will help physicians in the diagnosis of colorectal cancer and treatment decisions.

1. Introduction

According to Global Cancer Statistics 2021 published in the official journal of the American Cancer Society (ACS), colorectal cancer is the 4th most common cancer and the 5th most common mortality, posing a serious threat to the health of the population [1, 2]. Another 2018 Cancer Registry Report by Health Promotion Administration reveals that colorectal cancer ranks 2nd in morbidity, 2nd in mortality, 1st in morbidity for men and 3rd for women, and 3rd in mortality for men and 4th for women [3].

In recent years, the incidence of colorectal cancer is on the rise year by year as a result of the changing living pattern, food culture, sedentary lifestyle, work environment, and other factors, even with the trend of increasing youthfulness. With the exclusion of family genetic history, the risk factors for colorectal cancer are associated with the poor habits of people's lives, irregular work habits, physical inactivity, work strain, lack of dietary control, and aging. It is also noted that in terms of the risk of colorectal cancer, ranging from age and genetic to environmental and lifestyle choices, factors such as obesity, low physical activity, active and passive smoking, and high salt and red meat consumption are correlated to a higher risk of colorectal cancer [4–6]. Regarding the impact of age, a recent study demonstrated a steady yearly increase in the risk of young-onset colorectal cancer [7].

For early detection of colorectal cancer, carbohydrate antigen 19-9 (CA 19-9) and carcinoembryonic antigen (CEA) are commonly used biomarkers [8]. There is a strong correlation between CA 19-9 and CEA levels in colorectal cancer patients ($p = 0.001$ and $p < 0.0001$, respectively), both of which are important biomarkers in the progression of colorectal cancer [9]. Persistent smoking is known to alter the prognostic value of postoperative serum CEA levels in colorectal cancer patients because smoking can increase serum CEA levels independent of the disease status [10].

Another recent breakthrough in the diagnosis of colorectal cancer is deep neural network visualization [11–14]. An image analysis method based on deep learning can not only accurately classify different types of polyps in the whole slide image but also generate the main areas and features on slides through the model visualization method. This visualization method could significantly reduce the cognitive burden of clinicians [11]. In recent years, the convolutional neural network (CNN) model has been applied in the relevant medical literature. The accuracy of most model validation from previous studies [12–14] related to colorectal cancer falls between 75.1% and 83.9%. Based on the result of this study, the accuracy of the tumor profile by colonoscopy, that is, 81.6%, is more precise than those, 75.7% and 75.1% respectively, of colorectal cancer tumor analysis studies in references [12, 13]. It is slightly lower than that, 83.9%, in reference [14]. However, the source of image acquisition and research method are different. The cost is relatively high and is seldom employed. The proximity and complexity of the organs in the human body, the image resolution, size, and angle can affect the accuracy of identifying the targeted tissues and lesions using the training model.

The biggest difference between CNN and multilayer perceptron (MLP) lies in the additional convolution layer and pooling layer. These two layers enable CNN to have the capability in extracting details from image or speech features, instead of simply extracting data for calculation like other neural networks [15, 16].

The fuzzy theory was introduced by Lotfi [17]. Then, fuzzy logic of the concept of linguistic variables is proposed in reference [18]. Nowadays, fuzzy systems are applied in different fields, such as household appliances, industrial system control, and image recognition [19]. The research also pointed out that the application of professional fuzzy rules could help in detecting colorectal cancer and help doctors to easily identify diseases [20].

In this study, the convolutional neural network (CNN) was used for the training and learning of feature extraction from colonoscopy images. According to the health level, the training data were classified into healthiness, benign tumors, and malignant tumors. Colonoscopy images in the three categories were randomly selected as test data, and designated case images were used to assess the similarity of tumor profiles between the designated image and the image from test data.

The practical results of this study are summarized in the following four points:

(1) By analyzing the polyp profile in colonoscopy images, the results can be used as a reference for physicians to diagnose the symptoms as well as increase the detection efficiency and reduce the misdiagnosis rate.

(2) The patient's physical data are combined with the risk of tumor for assessment in the fuzzy system, which not only allows the patient to understand his current physical condition through data analysis but also enables the physician to make corresponding treatment decisions for the patient through the assessment results.

(3) After discussion on the results with clinicians, the accuracy of the raw data and assessment results are consistent with clinical analysis.

(4) The accuracy of the results of this study reaches 81.6%. Compared with the accuracy of colorectal cancer tumor analysis and research in the literature, the accuracy is better [12, 13], and the accuracy of the original data and evaluation results after discussion with clinicians is in line with clinical analysis.

2. Methodology

This study is based on the case data of colorectal cancer in a medical center in southern Taiwan. The colonoscopy images are trained and learned by convolutional neural networks. After completing the learning verification, the colonoscopy images of the designated patients will be tested and identified. Finally, the severity score and the patient-related information are fuzzy analyzed through a fuzzy algorithm, and the output is the risk of the patient's colorectal cancer, so that the doctor can diagnose colorectal cancer-related diseases Time aids.

3. Methods

Between January 2016 and December 2020, the medical records of the first 500 adults (i.e., age >18 years) of both genders undergoing first-time colonoscopy at a single referral center (i.e., Kaohsiung Chang Gung Memorial Hospital) regardless of indications were retrospectively reviewed. Exclusion criteria were as follows: (1) patients receiving previous colonoscopic examinations at other medical institutes, (2) those with normal colonoscopic findings, (3) those with a known history of benign or malignant colorectal diseases including familial polyposis and inflammatory bowel disease (i.e., Crohn's disease and ulcerative colitis), (4) those having received colorectal procedures (e.g., polypectomy and colorectal resection), (5) those without pathological analysis of colorectal specimens, and (6) those without complete information for the present study (e.g., body mass index and circulating CEA levels). Circulating CEA levels were determined in participants of annual physical checkups and those with positive stool occult blood test scheduled for colonoscopy.

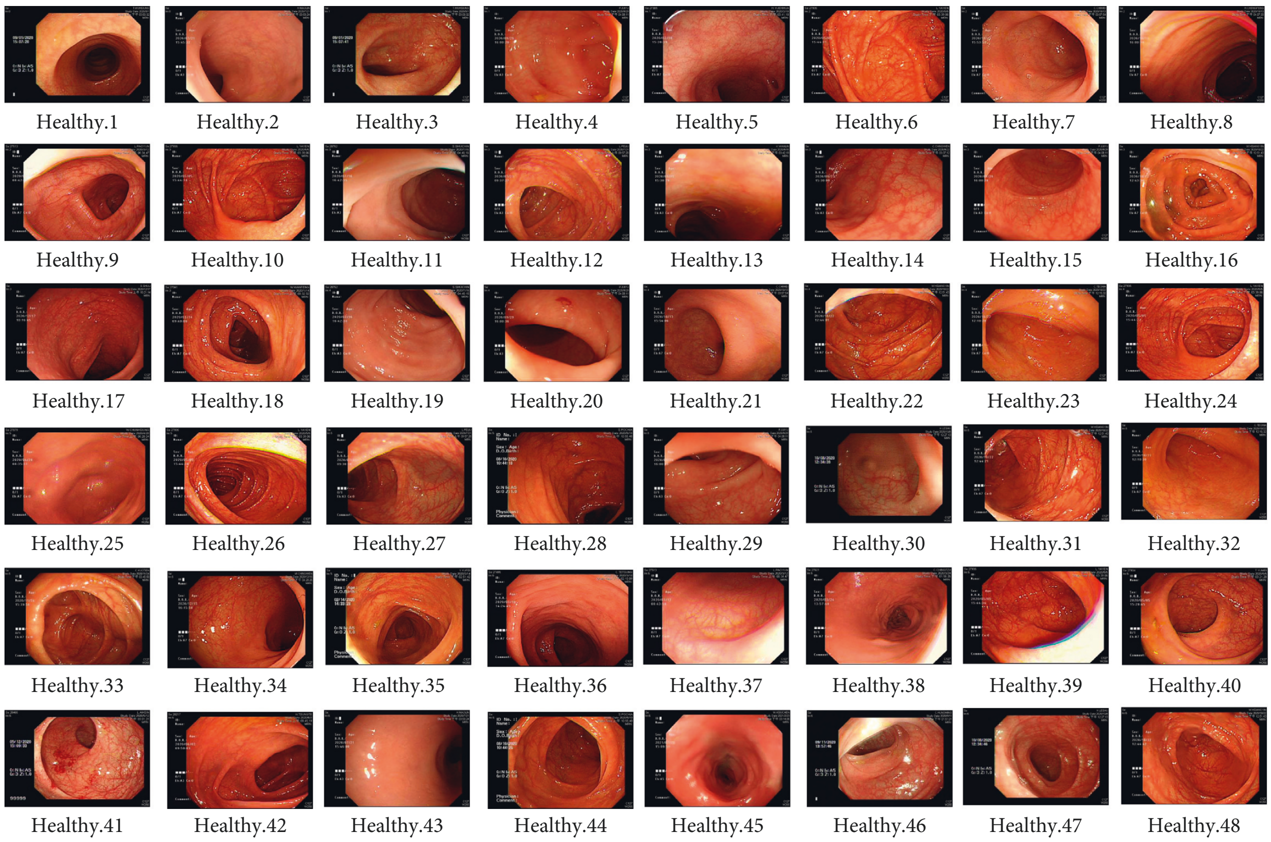


Figure 1: Healthy (normal).

4. Results

Of the 992 adult patients receiving colonoscopic examination within the study period, the medical records of the first 500 eligible for the current study were reviewed. The patient population comprised 275 males (55%) and 225 females (45%) with a mean age of 62.1 ± 11.8 (range, 31–85), a mean body mass index of 23.5 ± 3.7 (range, 17.5–31.6), and a mean circulating level of CEA 23.5 ± 280.6 (range, 0.5–1212.0).

In our routine practice, we take 12 images from a patient during a colonoscopic examination. Therefore, we had a total of 5950 images from 500 patients. All images were fed into an imaging analyzing software (Spyder 4.2.0) that divided the images into three categories, namely, normal image (Figure 1), benign (Figure 2), and malignant (Figure 3) tumors. There was no human handling or annotation of the images in the analytic process.

4.1. Research Environment. The colonoscopy images of 500 cases were divided into a training set, validation set, and test set. Among them, the colonoscopy images of 10 cases in the total data are taken as the test set, the rest is taken as the training set, and the images from the training set of 10% are used as the validation set of the training model for cross validation. Table 1 lists the environment and hardware configuration of this research.

4.2. System Software Design and Composition. Two kinds of software are used in the development of the system, one is Spyder, which is based on the development environment under the Python language. The open-source cross-platform scientific computing integrated development environment (IDE) of the Python language provides advanced code editing, interactive testing and debugging, computational science, data processing, and predictive analysis and supports multiple programming languages and operating systems. The second is Matlab, which is an interactive development environment based on algorithm development, data analysis, and numerical calculation.

4.3. Experimental Steps

(1) Obtain the image data of the colonoscopy through a medical center in the south, and search for the body-related data of the case based on the image data.

(2) Keep the required data, delete the unnecessary data such as blurred images, overexposed images, and unrecognizable shooting angles during the colonoscopy, and classify them into healthy (normal), benign, and malignant according to the type and appearance characteristics of polyps.

(3) From the three types of image data, after the training set and the test set are separated, the convolutional neural network is used to learn and train them, and the parameter values are adjusted to make the verification accuracy and loss value reach the expected value. Set goals.

Figure 2: Benign tumor.

(4) After completing the learning and training, perform the result verification to check the corresponding similarity evaluation of healthy (normal), benign, and malignant in the test images. This value is also used as the severity percentage in the fuzzy input, which is called the tumor risk.

(5) Finally, input the age, BMI value, tumor risk, and carcinoembryonic antigen index in the body data of the corresponding case into the fuzzy algorithm to evaluate the risk of cancer, and display the result.

4.4. Data Preprocessing. In the process of colonoscopy, the large intestine will be affected by factors such as the width of the intestine, bending of the intestine, the number of folds in the intestinal wall, the position and size of polyps, the shooting angle of the lens and whether it is accurately focused, and the overexposure or insufficient light source. The output image quality of the mirror inspection, coupled with the limited time of the inspection process, makes it inevitable that there will be poor quality and difficulty to identify image data in the screening results. Therefore, after filtering them, the difficult-to-identify or poor-quality images are deleted to improve the accuracy of the training model.

Since the human intestine is very long and the affected part only exists in a certain part of the general intestine, the results of colonoscopy screening may include normal images (labeled as "healthy") as well as those of polyps and malignant tumors. The colonoscopy images of these 500 patients are classified into three categories: "healthy" (Figure 1), "benign (Figure 2)," and "malignant" (Figure 3). This classification is only based on the appearance of polyps as a preliminary assessment, and the final judgment of the tumor profile must be approved by a professional physician. Screening and diagnosis are performed.

4.5. Convolutional Neural Network Model Architecture and Parameter Settings. The neural network model used in this research is SmallerVGGNet, which is the simplified CNN model architecture of VGGNet [21], and the colonoscopy data of 500 cases were classified into healthy (normal), benign, and malignant categories. In order to avoid overfitting of CNN during training, the database was divided into a training set and a test set without duplication, and 5%–10% of the images in the training set were taken as the validation set, which was repeatable. The purpose was to observe the validation accuracy of the model after training and select the training model with the highest validation accuracy as the CNN model in this study for tumor risk assessment.

The CNN architecture of this study is based on the SmallerVGGNet neural network as a multiconvolutional deep learning classifier, which consists of 7 convolutional layers and 4 pooling layers with MaxPooling added after convolutional layers 1, 3, 5, and 7, respectively. The remaining model parameters are presented in Table 2, with the activation function being ReLU in the CNN training model, sigmoid in the multitag classifier, adam in the

Figure 3: Malignant tumor.

Table 1: Experimental setup.

Test system	
Operating system	Windows 10 Professional (x64)
CPU	Intel Core i5-4570 (4C4T@3.2 GHz)
GPU	NVIDIA GeForce GT-710 2 GB
Memory	16 GB DDR3 1333 MHz
Development environment	Tensorflow-Keras (Spyder 4.2.0) Matlab R2020a
Program language	Python 3.6
Training set/test set	5950 piece/225 piece

Table 2: CNN model parameter settings.

Various parameters	
Layers	7 layers of convolutional layer and 4 layers of pooling layer
Convolutional kernel	3×3
MaxPooling	2×2
Activation function	Multilabel classification: sigmoid neural network training model: ReLU
Optimizer	Adam
Stride	1
Dropout	25%
Learning rate	0.001
Batch size	32
Epoch	200
Number of hidden layer neurons	1024

optimizer, stride of 1, dropout of 25%, the initial learning rate of 1e-3, batch size of 32, training iteration epoch of 200, and hidden layer neurons of 1024.

4.6. Fuzzy System Design. The tumor risk estimated by the CNN training model can be combined with the patient's body-related data to derive the risk of colorectal cancer. Therefore, a fuzzy system was designed by establishing semantic variables of input and output, defining their membership functions, and formulating fuzzy rules, fuzzy inference, and defuzzification. As such, a fuzzy system for the risk of cancer was determined.

The tumor risk derived from the CNN model was combined with four input variables, including the corresponding age of the patients, BMI, and carcinoembryonic antigen. After fuzzification of the triangular and trapezoidal membership functions, the maximum-minimum (max-min) synthesis operator was used for the computation of the membership of the fuzzy set by the center-of-gravity method and the output was the risk of colorectal cancer.

4.6.1. Design of Fuzzy Parameters. The fuzzy algorithm was applied to assess the risk of colorectal cancer, as shown in Figure 4, The tumor risk derived from the CNN model was combined with four input variables, including the corresponding age of patients, BMI, tumor risk, and

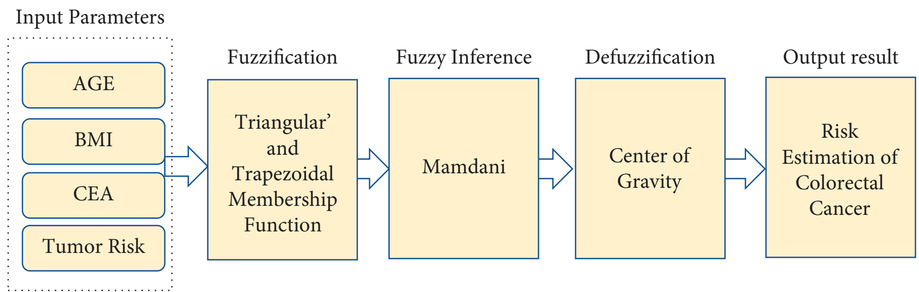


Figure 4: Schematic diagram of fuzzy evaluation.

Risk Indicator	Years
Very low risk group (VL)	0~20
Low risk group (L)	15~40
Low-medium risk group (LM)	30~50
Middle-risk group (M)	40~60
High-Medium risk group (HM)	50~70
High risk group (H)	60~85
Very High RiskGroup (VH)	80

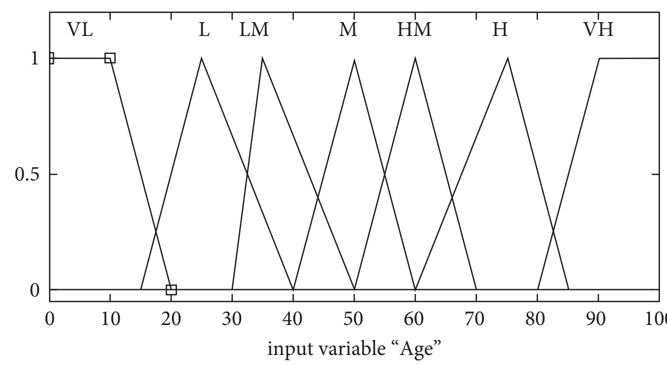


Figure 5: Input attribution function design-age risk index.

Health Indicators	BMI
Underweight	0~18.5
Normal	16~25
Overweight	22~29
1 Degree Fat	25~32
2 Degree Fat	29~40
3 Degree Fat	35

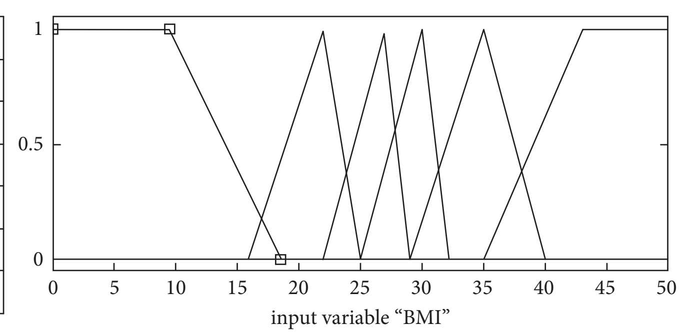


Figure 6: Input attribution function body mass index, BMI.

Reference Indicator	CEA (ng/mL)
Normal	0~3.5
Smoker	1.5~7
Abnormal	5 above

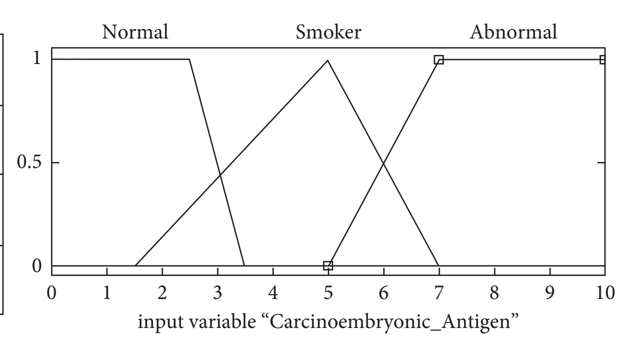


Figure 7: Input attribution function carcinoembryonic antigen, CEA.

carcinoembryonic antigen. After fuzzification of the triangular and trapezoidal membership functions, the maximum-minimum (max-min) synthesis operator was used for the computation of the membership of the fuzzy set by the center-of-gravity method and the output was the risk of colorectal cancer.

Severity Rating	Risk of Tumor (%)
Healthy(Normal)	0~40
Benign	25~75
Malignant	55~100

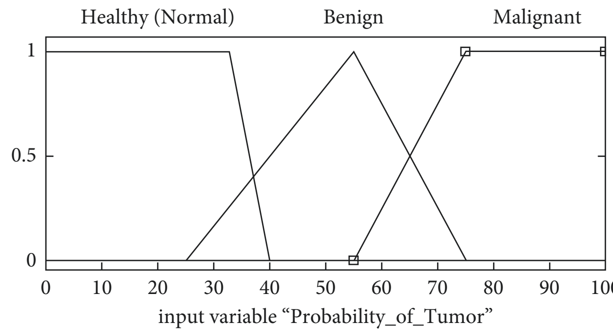


Figure 8: Input attribution function risk of tumor.

Risk Iindicator	Risk of Tumor (%)
Low-risk group(LR)	0<×≦20
Low-to-medium risk(LM)	20<×≦40
Medium risk group(MR)	40<×≦60
High-medium risk group(HM)	60<×≦80
High-risk group(HR)	×>80

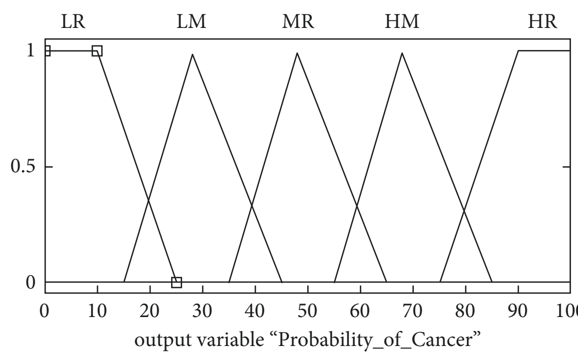


Figure 9: Risk of cancer.

Table 3: Risk of tumor. (a) Healthy: age/BMI; (b) age/CEA; (c) CEA/BMI.

(a) Healthy: age/BMI

BMI age (years)	0–20	15–40	30–50	40–60	50–70	60–85	Over 80
Underweight	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk
Normal weight	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk
Overweight	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk
Severe obesity	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk
Morbid obesity	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk
Super obesity	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk

(b) Healthy: age/CEA

CEA age (years)	0–20	15–40	30–50	40–60	50–70	60–85	Over 80
Normal	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk
Smoker	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk
Abnormal	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk	Low risk

(c) Healthy: CEA/BMI

BMI age	Normal	Smoker	Abnormal
Underweight	Low risk	Low risk	Low risk
Normal weight	Low risk	Low risk	Low risk
Overweight	Low risk	Low risk	Low risk
Severe obesity	Low risk	Low risk	Low risk
Morbid obesity	Low risk	Low risk	Low risk
Super obesity	Low risk	Low risk	Low risk

4.6.2. Establishment of Semantic Variables and Membership Functions. The fuzzy system has four input parameters and one output. Input parameters are age in Figure 5, BMI in Figure 6, carcinoembryonic antigen in Figure 7, and tumor risk in Figure 8, respectively, and the output is risk of colorectal cancer in Figure 9 The terms and the membership functions of each parameter are explained below.

Once the terms were established, the membership functions were defined based on the data from the pieces of literature. The graphs of membership function of this study were based on the triangular (Trimf) and trapezoidal (Trapmf) membership functions referenced to obtain better results and to facilitate the observation of the data in this study, while the range of age membership was determined

Table 4: Risk of tumor. (a) Benign: age/BMI; (b) age/CEA; (c) CEA/BMI.

(a) Benign: age/BMI

BMI age (years)	0–20	15–40	30–50	40–60	50–70	60–85	Over 80
Underweight	Low risk	Low risk	Low risk	Low-to-moderate risk	Moderate risk	Moderate risk	Moderate risk
Normal weight	Low risk	Low risk	Low risk	Low-to-moderate risk	Moderate risk	Moderate risk	Moderate risk
Overweight	Low risk	Low risk	Low-to-moderate risk	Moderate risk	Moderate risk	Moderate-to-high risk	Moderate-to-high risk
Severe obesity	Low risk	Low risk	Low to moderate risk	Moderate risk	Moderate-to-high risk	Moderate-to-high risk	High risk
Morbid obesity	Low risk	Low-to-moderate risk	Low-to-moderate risk	Moderate risk	Moderate-to-high risk	High risk	High risk
Super obesity	Low risk	Low-to-moderate risk	Low-to-moderate risk	Moderate-to-high risk	High risk	High risk	High risk

(b) Benign: age/CEA

CEA age (years)	0–20	15–40	30–50	40–60	50–70	60–85	Over 80
Normal	Low risk	Low risk	Low risk	Low-to-moderate risk	Moderate risk	Moderate risk	Moderate-to-high risk
Smoker	Low risk	Low risk	Low-to-moderate risk	Moderate risk	Moderate risk	Moderate-to-high risk	High risk
Abnormal	Low risk	Low-to-moderate risk	Low-to-moderate risk	Moderate risk	Moderate-to-high risk	High risk	High risk

(c) Benign: CEA/BMI

BMI age	Normal	Smoker	Abnormal
Underweight	Low risk	Low risk	Low risk
Normal weight	Low risk	Low risk	Low-to-moderate risk
Overweight	Low risk	Low-to-moderate risk	Moderate risk
Severe obesity	Low-to-moderate risk	Moderate risk	Moderate-to-high risk
Morbid obesity	Moderate risk	Moderate-to-high risk	High risk
Super obesity	Moderate-to-high risk	High risk	High risk

Table 5: Risk of tumor. (a) Malignant: age/BMI; (b) age/CEA; (c) CEA/BMI.

(a) Malignant: age/BMI

BMI age (years)	0–20	15–40	30–50	40–60	50–70	60–85	Over 80
Underweight	High risk	High risk	High risk	High risk	High risk	High risk	High risk
Normal weight	High risk	High risk	High risk	High risk	High risk	High risk	High risk
Overweight	High risk	High risk	High risk	High risk	High risk	High risk	High risk
Severe obesity	High risk	High risk	High risk	High risk	High risk	High risk	High risk
Morbid obesity	High risk	High risk	High risk	High risk	High risk	High risk	High risk
Super obesity	High risk	High risk	High risk	High risk	High risk	High risk	High risk

(b) Malignant: age/CEA

CEA age (years)	0–20	15–40	30–50	40–60	50–70	60–85	Over 80
Normal	High risk	High risk	High risk	High risk	High risk	High risk	High risk
Smoker	High risk	High risk	High risk	High risk	High risk	High risk	High risk
Abnormal	High risk	High risk	High risk	High risk	High risk	High risk	High risk

(c) Malignant: CEA/BMI

BMI age	Normal	Smoker	Abnormal
Underweight	High risk	High risk	High risk
Normal weight	High risk	High risk	High risk
Overweight	High risk	High risk	High risk
Severe obesity	High risk	High risk	High risk
Morbid obesity	High risk	High risk	High risk
Super obesity	High risk	High risk	High risk

from the statistics of reference [11], and BMI and CEA obtained from the information from the Ministry of Health and Welfare and major hospitals.

4.6.3. Establish Fuzzy Rule Base. After evaluating the polyp profile of colorectal endoscopy based on the CNN network model, combining the relevant risk factor parameters and the clinical experience of professional physicians, the corresponding results can be summarized, which is also used as a reference for the design of the fuzzy rule library. In the rule table of the fuzzy system in this study, there are 7 semantic variables in "age," 6 semantic variables in "BMI value," 3 semantic variables in "carcinoembryonic antigen index," and 3 in "tumor risk." There are semantic variables, so there are a total of 378 rules. Tables 3–5 list the comparison of the fuzzy rule base of healthy, benign, and malignant.

4.6.4. Fuzzy Inference and Defuzzification. In the fuzzy system of this study, the method of the center of gravity was utilized for the computation of the defuzzification. With the center of gravity method, the tumor risk and the three risk factors can be derived to assess the risk of cancer. Finally, the probabilistic assessment of the risk level allows the physician to know the current physical data of the patient to assist the physician in the diagnosis and treatment process, thus increasing the efficiency of diagnosis and reducing the rate of misdiagnosis. In the case of a patient aged 75, with a BMI of 25.5, a CEA of 10.55, and a tumor risk of 60.5%, the risk of cancer is calculated to be 75.3%, which corresponds to a risk assessment of the "moderate-to-high risk group."

5. Results

5.1. The Proposed CNN Training Model. Following the consummation of design of the neural network architecture, the accuracy and loss function of the model were observed by varying the number of iterations and the ratio of images in the training and validation sets. The accuracy ranged from 0.6 to 0.65 for 50 and 75 iterations, indicating a poor training effect. Figure 10 shows the training results of the model with 100, 150, 175, and 200 iterations, and it can be therefore observed that when the ratio of the training set to the validation set was 9:1 for 200 iterations, the accuracy rate reached 81.6%, which was the model with the optimal training effect after multiple adjustments. Thus, it was chosen as the CNN training model for this study.

5.2. Analysis of the Risk of Tumor Detection by Colonoscopy. After the CNN training model was selected, the image data from the test set were classified and identified by a multi-convolutional classification Keras model. The results of the colonoscopy images in Figure 11 illustrate the percentage of the images in each of the three categories of healthy, benign, and malignant after assessment by the classification model, and the assessed probabilities were benign, 87.88%; malignant, 24.19%; healthy, 0.02%. Since the three categories of healthy, benign, and malignant were analyzed separately in the assessment process, the results were not 100% for the three categories combined; instead, the percentages of the three categories were assessed separately for each image. From the above assessment results, the risk of the polyp profile being a benign tumor was 87.88%.

5.3. Assessment of the Risk of Colorectal Cancer. There were four input parameters of the fuzzy system in this study, among which tumor risk was estimated by the CNN model, and the remaining age, BMI, and CEA indexes were the physical data of the patient corresponding to the colonoscopy images. The risk of colorectal cancer was measured by the fuzzy system, as shown in Figure 12 the result of

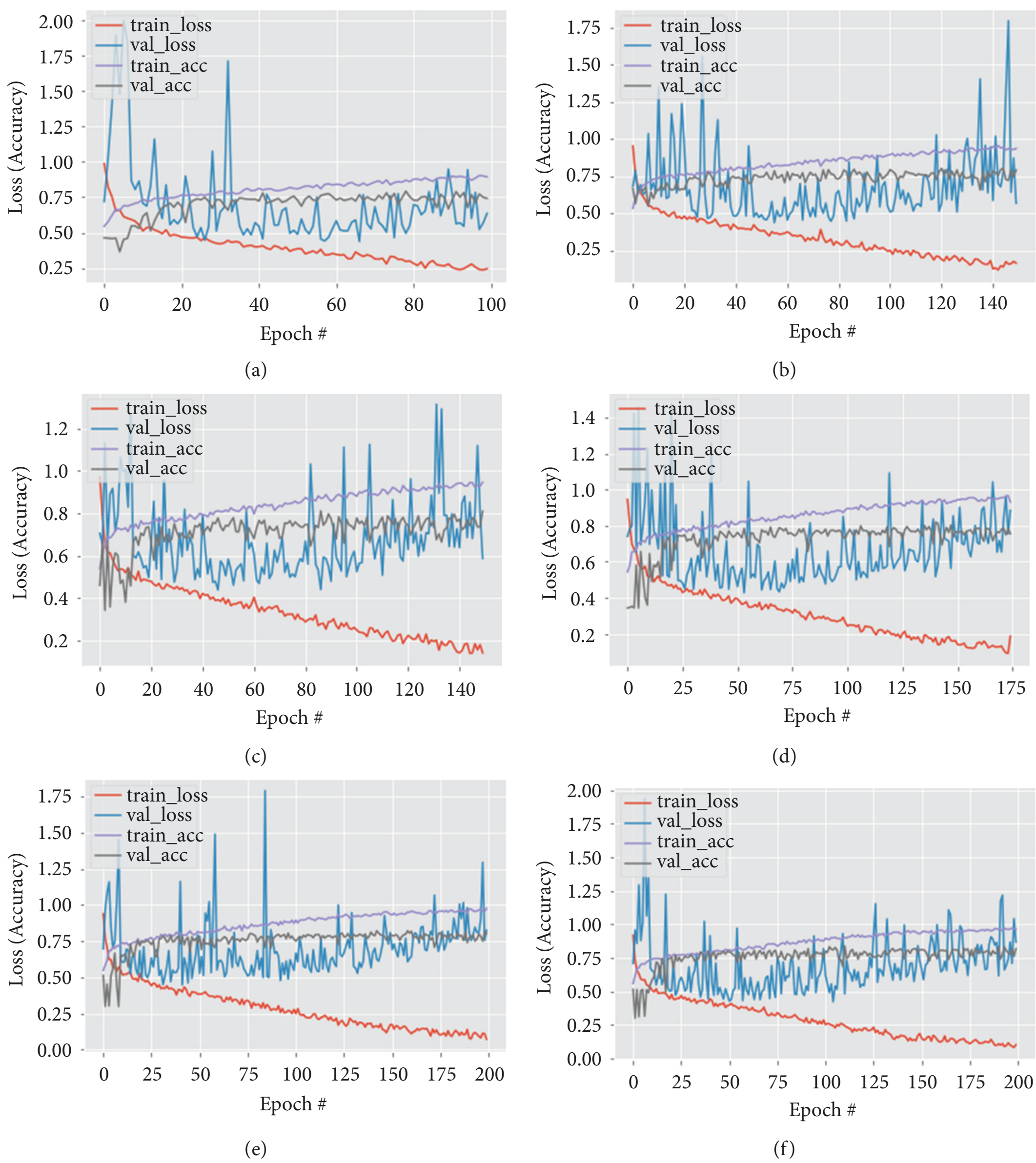


Figure 10: Epochs and validation accuracy difference comparison. (a) Epochs: 100, validation accuracy: 0.7424. (b) Epochs: 150, validation accuracy: 0.7918. (c) Epochs: 150, validation accuracy: 0.8108. (d) Epochs: 175, validation accuracy: 0.7555. (e) Epochs: 200, validation accuracy: 0.8027. (f) Epochs: 200, validation accuracy: 0.8161.

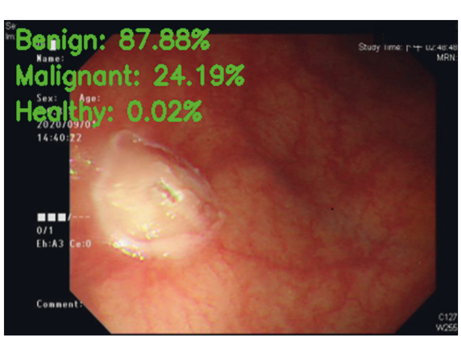


Figure 11: Tumor risk analysis result: the risk of benign tumor is 87.88%.

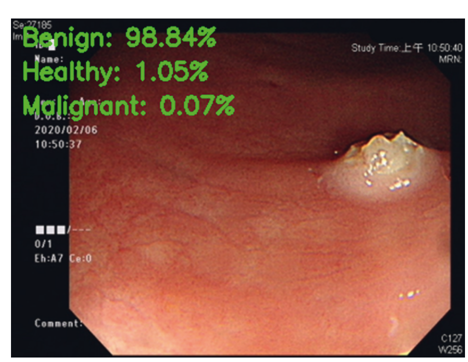


Figure 12: Tumor risk assessment result: the risk of benign tumor is 98.84%.

colonoscopy image assessment was as follows: the risk of benign tumor is 98.84% and is 65.23% after the conversion of severity level, while the age of the patient was 44 years old, with a BMI of 31.6 and a CEA index of 2.01, and the risk of colorectal cancer was 60.6% after the assessment by the fuzzy system.

6. Discussion

The results of this study are intended for diagnostic purposes, not as a substitute for a physician's decision, thus there is no definite accuracy rate. Apart from discussing the results of the analysis of different tumor probabilities and physical conditions, the cases are also observed to see whether the accuracy rates are clinically relevant in various situations.

Case 1. Age = 56, BMI = 20.1, CEA = 5.15, PoT = 90.71%, and the fuzzy system assesses the risk of cancer as 91.7%, which corresponds to the interval of high-risk group. The risk factors in this case are all within normal values except for the age risk, which indicates that although this study adopts relatively stable risk factors as the experimental data, all risk factors can only function to the extent of increasing or decreasing the risk, and the actual physical condition has to be properly assessed by screening.

Case 2. Age = 61, BMI = 26.8, CEA = 1.09, PoT = 64.05%, and the fuzzy system assesses the risk of cancer as 60.2%, which corresponds to the interval of medium-high risk group. The possibility of cancer in this case is subject to assessment by the physician.

Case 3. Age = 47, BMI = 24.9, CEA = 434.78, PoT = 25.7%, and the fuzzy system assesses the risk of cancer as 10.1%, which corresponds to the interval of low-risk group. This case shows an unusually high CEA. Although CEA is only a reference value for risk factors, neither a high value means cancer, nor a low value means no cancer. As the human intestine is very long and the CEA in this case is significantly elevated above the abnormal range, this phenomenon suggests that the lesion may be located elsewhere in the intestine.

Taking into account the promising association between the imaging outcomes and the results of pathological analyses, the current study highlighted a time-efficient and noninvasive approach to the diagnosis of potential colorectal malignancies. The imaging tool may provide clinical guidance for clinicians to determine whether to proceed with high-risk procedures (e.g., polypectomy) or adopt a more conservative strategy, particularly in patients at high risk of complication (e.g., coagulopathy or impending colon perforation). Another advantage is the lack of requirement for specific software for operation. Nevertheless, the current study has its limitations. First, despite the involvement of up to 500 patients in the current study, the sample size is still relatively small to consolidate our findings. Second, although we included patient factors including age, CEA, and body mass index in our analysis, other risk factors for colorectal cancers such as dietary habits and family history were not taken into account. Further large-scale studies are warranted to validate the clinical application of the present imaging approach. Finally, the lack of real-time feedback is another limitation. Nevertheless, analyses of all images from a single patient can be completed within five minutes. In routine practice, such a short time could allow a clinician to decide the appropriate colonoscopic management strategy (i.e., invasive vs. conservative) based on the results of analysis when the patient is still under anesthesia or sedation.

The lack of real-time feedback is another limitation. Nevertheless, analyses of all images from a single patient can be completed within five minutes. In routine practice, such a short time could allow a clinician to decide the appropriate colonoscopic management strategy (i.e., invasive vs. conservative) based on the results of analysis when the patient is still under anesthesia or sedation.

Essentially the fuzzy rules based on the way-wise physicians' design and a large amount of learning information will reduce, but not completely avoid, the misdiagnosis rate.

7. Conclusion

In this study, a CNN training model is employed to analyze the assessment of tumor risk of healthy, benign, and malignant on colonoscopy images, and then the four parameters of tumor risk, age, BMI, and CEA are utilized to assess the risk of colorectal cancer by fuzzy algorithm, which assists physicians to effectively diagnose patients' symptoms through their current physical condition and data, thus reducing the misdiagnosis rate.

However, from the analysis and discussion of the experimental results, it is evident that despite the high priority of assessed tumor risk in the fuzzy system, as colonoscopy is the most direct way to screen for colorectal cancer, the consideration of risk factors has a certain degree of reference value for diagnostic signs, clinical analysis, postoperative follow-up, and prevention, in addition to the current physical condition.

Among the many colorectal cancer screening methods, colonoscopy is currently the most important and direct screening method. Compared with other methods, this method can directly observe the general situation of all tumors in the intestine. Compared with medical image computing together with MICCAI, CVC Colon DB, and ISIT-UMR, the Association for Computer-Aided Intervention used the image sequence data in the medical dataset as the training model of the deep convolutional neural network (DCNN) and trained two sets of settings Set-1 and DCNN. The accuracy rates of Set-2 are 75.71% and 79.78%, respectively [12]; based on computer-aided diagnosis (CAD), combined with convolutional neural network (CNN), through deep learning model after training, the polyp status analyzed by CAD in colonoscopy was used for verification test with CNN, and the accuracy rate of the research results was 75.1% [13].

The proximity and complexity of the organs in the human body, the image resolution, size, and angle can affect the accuracy of identifying the targeted tissues and lesions using the training model. The biggest difference between CNN and multilayer perceptron (MLP) lies in the additional convolution layer and pooling layer. According to the Cancer Registration Annual Report of the National Health Administration of the Ministry of Health and Welfare of Taiwan, 17,302 people were initially diagnosed with colorectal cancer in 2019, of which 11,031 (63.8%) were colon cancer, and 6,271 (36.2%) were rectal, sigmoid junction, and anus [3]. Future research will be directed towards designing CNN network model to observe the difference in the diagnostic accuracy of colorectal cancer in different parts.

Data Availability

Access to data is restricted and the data are not freely available. Acceptable justifications for restricting access may include legal and ethical concerns, such as third-party rights, patient privacy, and commercial confidentiality.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research work was partly supported by the Ministry of Science and Technology, ROC, Grant no. MOST 110-2221-E-992-093.

References

[1] S. Hyuna, F. Jacques, L. Rebecca et al., "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *Ca - a Cancer Journal for Clinicians*, vol. 70, pp. 145–164, 2020.

[2] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2017," *CA: A Cancer Journal for Clinicians*, vol. 67, no. 1, pp. 7–30, 2017.

[3] *Cancer Registry Annual Report 2018*, pp. 1–5, Taiwan health promotion administration Ministry of health and Welfare, Taipei, Taiwan, 2020.

[4] L. Anna, R. Grzegorz, L. Tomasz, S. G. Aleksandra, and R. Sławomir, "Risk factors for the diagnosis of colorectal cancer," *Research Square*, vol. 29, pp. 1–15.

[5] A. H. Fatima and P. B. Robin, "Colorectal cancer epidemiology: incidence, mortality, survival, and risk factors," *Clinics in Colon and Rectal Surgery*, vol. 22, pp. 191–197, 2009.

[6] C. C. Kuan, C. L. Ko, H. C. Hong, C. C. Kung, L. W. Kuen, and C. S. Ling, "Path analysis of the impact of obesity on postoperative outcomes in colorectal cancer patients: a population-based study," *Journal of Clinical Medicine*, vol. 10, pp. 1–11, 2021.

[7] H. Alyssa, C. S. Eric, M. L. Jonathan et al., "Trends in the incidence of young-onset colorectal cancer with a focus on years approaching screening age: a population-based longitudinal study," *JNCI J Natl Cancer Inst*, vol. 113, pp. 863–868, 2021.

[8] A. Subki, A. ButtNS, and A. A. Alkahtani, "CEA and CA19-9 levels and KRAS mutation status as biomarkers for colorectal cancer," *Clinical Oncology*, vol. 6, pp. 1–8, 2021.

[9] L. Leilani, S. Silvia, W. Mathias, H. B. Doris, K. Marko, and L. Johannes, "Diagnostic and prognostic value of CEA and CA19-9 in colorectal cancer," *Diseases*, pp. 1–12, 2021.

[10] C. S. Huang, C. Y. Chen, L. K. Huang, W. S. Wang, and S. H. Yang, "Prognostic value of postoperative serum carcinoembryonic antigen levels in colorectal cancer patients who smoke," *PLoS One*, vol. 6, no. 5, pp. 1–14, 2020.

[11] K. Bruno, M. O. Andrea, P. M. Allen, M. N. Catherine, A. S. Matthew, and T. Lorenzo, "Looking under the hood: deep neural network visualization to interpret whole-slide image analysis outcomes for colorectal polyps," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern RecognitionWorkshops*, Honolulu, HI, USA, July 2017.

[12] P. Krushi, L. Kaidong, T. Ke et al., "A comparative study on polyp classification using convolutional neural networks," *PLoS One*, vol. 7, no. 30, pp. 1–16, 2020.

[13] K. Yoriaki, H. Hisashi, W. Tomohiro et al., "Computer-aided diagnosis based on convolutional neural network system for colorectal polyp classification: preliminary experience," *Oncology*, vol. 93, pp. 30–34, 2017.

[14] Y. Atsuo, N. Ryota, O. Keita, A. Tomonori, and K. Kazuhiko, "Automatic detection of colorectal neoplasia in wireless colon capsule endoscopic images using a deep convolutional neural network," *Endoscopy*, vol. 53, pp. 832–836, 2021.

[15] W. Liuli, Z. M. Liu, and Z. T. Huang, "Deep convolution network for direction of arrival estimation with sparse prior," *IEEE Signal Processing Letters*, vol. 26, pp. 1688–1692, 2019.

[16] D. D. Pukale, S. G. Bhirud, and V. D. Katkar, "Content based image retrieval using deep convolution eural network," *IEEE Xplore*, pp. 1–5, 2017.

[17] A. Z. Lotfi, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[18] A. Z. Lotfi, "Outline of a new approach to the analysis of complex systems and decision processes," *Man and Cybernetics*, vol. 3, pp. 28–44, 1973.

[19] S. K. Halgamuge and M. Glesner, "Neural networks in designing fuzzy systems for real world applications," *Fuzzy Sets and Systems*, vol. 65, no. 1, pp. 1–12, 1994.

[20] C. Tanjia, "Fuzzy logic based expert system for detecting colorectal cancer," *International Research Journal of Engineering and Technology*, vol. 05, pp. 389–393, 2018.

[21] S. Karen and Z. Andrew, "Very deep comvolutional networks for large-scale image recognition," in *Proceedings of the Conference Paper at International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 2508568, 11 pages
https://doi.org/10.1155/2022/2508568

Research Article

Control for Isokinetic Exercise with External Disturbance

Xinwei Guo,[1] **Yixuan Sheng**,[2] **Jinbao Liu**,[2] **and Honghai Liu**[2]

[1]*Department of Industrial Design, School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100084, China*
[2]*State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

Correspondence should be addressed to Honghai Liu; honghai.liu@icloud.com

Received 24 February 2022; Accepted 16 March 2022; Published 9 May 2022

Academic Editor: Bin Xu

Isokinetic exercise is considered as one of the most effective ways for rehabilitation and muscle strength enhancement. As rehabilitation equipment, the isokinetic exercise device assists isokinetic movement using a single fixed axis and relevant joint adapters, and it requires high stability and safety performances to cooperate with human. In this paper, a brushless DC motor (BLDCM) was adopted to make a study of isokinetic control. Fractional-order fuzzy PID (FOFPID) controller is an effective controller for the nonlinear system to realize isokinetic control according to reference speed and irregular disturbance (load torque). The model of the BLDCM and FOFPID controller system is built in MATLAB/Simulink to simulate the velocity response of the controller, and a comparison between the FOFPID controller and PID controller is made to verify the system stability.

1. Introduction

Relevant research studies of muscular strength enhancement are extremely meaningful for sports and rehabilitation areas to improve the competitive performance and physical fitness [1, 2]. Some isokinetic, isometric, and isotonic physical training could be used with the equipment to achieve the ideal effect of overcoming resistance to exercise muscle strength [3–5]. According to some research studies of muscle contractions, isokinetic exercise is considered as the safest and reliable muscular exercise mode; meanwhile, isokinetic dynamometry is regarded as one of the most credible muscle strength assessments [6]. Hence, isokinetic exercise is increasingly applied for people of dyskinesia and athletes to recover motion ability or enhance muscular strength. To maintain a constant velocity, the isokinetic exercise is accomplished with the assistance of an isokinetic exercise device. At present, isokinetic exercise device mainly aims at “single-joint” configurations which drive a single fixed axis with different adapters corresponding to each joint assembling on it. The joint angular velocity is constant and preset during isokinetic movement while the resistance moment provided by the isokinetic device is equal to joint torque and changes following joint torque [7]. Because of the resistance moment and reciprocating motion provided by the isokinetic exercise device, the isokinetic control under external torque disturbance needs to meet high robustness and reliability performance to achieve stability requirement and security interaction with experimenters. As one of the rehabilitation instruments, isokinetic exercise device has a broad development prospect and potential for hemiplegic patients and athletes. This paper mainly considered the study of isokinetic control to enhance the stability and lay the foundation of isokinetic exercise device development.

Direct current motor (DC motor) has been widely used in industries for several years, owing to its advantages of stability, fast response, high efficiency, simple control system, etc. Brushless DC motor (BLDCM) has evolved from DC motor driven from pulse width modulation (PWM) inverters which overcomes the disadvantages of speed, lifetime, and noise [8, 9]. For isokinetic control application, the system is nonlinear and BLDCM needs a more complicated controller to achieve constant speed control compared with traditional DC motor. Many researchers have dedicated themselves to study effective control strategies to increase the capability of speed tracking under disturbance and sudden torque change [10, 11]. As a simple and useful controller, conventional proportional-integral-derivative (PID) controller is applied in industrial control systems generally because of its performance of high adaptation and simplicity [12]. However, its low-precision, poor anti-jamming ability and fixed parameters make PID controller fail to satisfy the high-performance requirements in some industrial applications especially for the systems with high disturbances [13]. Therefore, various speed controllers have been studied on the basis of conventional and intelligent control strategies. Xia et al. discussed switching-gain adaptation current control to achieve high performance with changing disturbance in dynamic and static processes. The result showed that the general disturbance could be eliminated in different operating stages and the controller improved the steady-state accuracy effectively [14]. Mandel and Weiss used a resonant controller in which resonant frequencies could be adjusted on the basis of motor speed to reduce torque ripple. According to the experimental tests, the controller showed the ability of tracking variable speed command and restraining the influence of cogging torque [15]. Wang et al. designed a quadratic single neuron (QSN) adaptive PID controller which could follow the reference speed successfully with sudden load disturbance [16]. Sivarani et al. proposed a novel bacterial foraging algorithm-optimized online adaptive neuro-fuzzy inference system (ANFIS) controller which performs better set point tracking and has better learning parameter tuning ability and time domain specifications [11]. Xu et al. proposed a novel real-time planning method for robot to simultaneously avoid obstacle and track the target [17]. For external disturbance, many researchers used appropriate control strategies according to their inherent properties to overcome the perturbation, such as active disturbance rejection control [18], neural network-based adaptive impedance control [19], repetitive control [20], and so on [21].

In terms of various intelligent control techniques, fuzzy logic control (FLC) is a robust, intelligent, and adaptive control strategy for complex nonlinear dynamic systems [22]. As the promotion of intelligent control strategies, FLC is developing towards the trend of self-adaptive with other controllers, such as fuzzy PID control [23], neuro-fuzzy control [24], fuzzy predictive control [25], and so on. Fractional-order fuzzy PID (FOFPID) controller is a combination of fuzzy controller and PID controller with noninteger orders in differential and integral parts. From some relevant research studies, the efficiency and robustness could be enhanced distinctly by combining fractional-order operator. Kumar and Rana applied nonlinear adaptive FOFPID controller for a 2-link planar rigid robotic manipulator with payload, and the results clearly revealed a good trajectory tracking performance [26]. Kumar et al. proposed a self-tuned robust FOFPID controller for a nonlinear active suspension system of a quarter car, and the simulation study showed a better performance by providing comfort ride during hard constraint [27]. Although the previous research studies of FOFPID presented better control capability, there are less research studies of isokinetic control with periodic velocities and irregular external load torque.

In this paper, a FOFPID controller was applied to control the isokinetic movement of BLDCM. Section 2 introduces the mathematical model of BLDCM and the simulation model in MATLAB/Simulink. FOFPID controller for isokinetic control is proposed in Section 3. Section 4 presents the simulation results and makes a comparison between FOFPID controller and PID controller. The discussion and conclusion are given in Section 5.

2. Model of BLDCM System

2.1. Mathematical Model of BLDCM System. The mathematical model of BLDCM system would be established under the condition of reasonable simplification. The following assumptions are proposed. (1) Stator has star connection, three-phase winding is completely symmetrical, and the model is working in two-phase conduction and three-phase six states. (2) The magnetic circuit of the motor is unsaturated during operation without eddy current and hysteresis loss. (3) The air gap is uniform and the magnetic field is a square wave. (4) The stator current and rotor magnetic field are symmetrical. (5) The armature windings are uniformly and continuously distributed on the inner surface of the stator. (6) The armature effect and cogging effect are ignored [28, 29].

Under the above assumptions, the voltage balance equation of three-phase winding could be obtained by the Kirchhoff voltage law (KVL):

$$\begin{gathered}\begin{bmatrix} u_A \\ u_B \\ u_C \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix}\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} L-M & & \\ & L-M & \\ & & L-M \end{bmatrix}\frac{d}{\mathrm{d}t}\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix}, \\ i_A + i_B + i_C = 0. \end{gathered} \tag{1}$$

where u_A, u_B, u_C refer to stator phase winding voltage (V); i_A, i_B, i_C refer to stator phase winding current (A); e_A, e_B, e_C refer to stator phase winding electromotive force (V); R is the motor phase resistance (Ω); L is self-inductance of each phase winding (H); and M is the mutual inductance between each two-phase winding (H).

The correlation between the stator winding input power and the generated electromagnetic torque is described by the following equation:

$$e_A i_A + e_B i_B + e_C i_C = T\boldsymbol{\omega}. \quad (2)$$

Hence, the electromagnetic torque of BLDCM could be formulated as follows:

$$T = \frac{(e_A i_A + e_B i_B + e_C i_C)}{\boldsymbol{\omega}}, \quad (3)$$

where T is the electromagnetic torque (Nm) and ω is the angular velocity of motor (°/s).

The electromagnetic torque of BLDCM is generated by the interaction between the current in the stator winding and the magnetic field generated by the rotor magnetic steel. From (3), it could be known that electromagnetic torque is proportional to the magnetic field and phase current and inversely proportional to the rotational angular velocity.

The motion equation could be expressed as

$$\mathrm{T} - T_L = J\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}, \quad (4)$$

where T_L is the load torque and J is the moment of inertia of motor.

For the isokinetic movement process, the motor angular velocity ω gradually increases during acceleration stage, then remains constant during isokinetic stage, and decreases during deceleration stage. The electromagnetic torque T (resistance moment) is equal to joint torque (load torque). Hence, the isokinetic motion state equations could be expressed as

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} = a > 0 \text{ acceleration stage} \\ \dfrac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} = 0 \text{ isokinetic stage} \\ \dfrac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} = -a < 0 \text{ deceleration stage} \\ T = T_L \end{cases}, \quad (5)$$

where ω is the preset joint angular velocity and a is the acceleration of acceleration and deceleration stages.

2.2. Modelling BLDCM in MATLAB/Simulink. The BLDCM system for isokinetic control consists of a BLDCM module, reference current module, PWM current controller module, voltage inverter module, and FOFPID controller module. The modular modelling schematic of the BLDCM system is shown in Figure 1. The system consists of two control parts: speed control and torque control. The FOFPID controller was applied for speed control, and a conventional PID controller was used for torque control. The application target of the BLDCM system is to simulate the isokinetic movement stage during isokinetic exercise with the preset speed and load torque by the experimenter. To validate the robustness and stability performance of the FOFPID controller and BLDCM system during isokinetic movement under disordered joint torque, the system model was built in MATLAB/Simulink as shown in Figure 2.

3. Isokinetic Control System of BLDCM

In this study, a fractional-order fuzzy PID controller, which combines the fractional-order calculus, fuzzy logic control, and conventional PID control, is proposed to control the speed of the BLDCM system for tracking the reference speed command under irregular load torque. The FOFPID controller has evolved from fuzzy PI and fuzzy PD controllers with self-tuning and adaptive capabilities [30]. The characteristic of the FOFPID controller is using noninteger order instead of integer order integrator and differentiator operators. In the subsequent sections, the design of the FOFPID controller is presented.

3.1. Fuzzy PI and Fuzzy PD Controllers. The basic structures of the FOFPID controller are fuzzy PI and fuzzy PD controllers, which could provide proportional, integral, and derivative parameters and fuzzy logic to realize adaptive control for the system. The algorithms of fuzzy PI and fuzzy PD controllers could be expressed as [26]

$$\begin{aligned} u_{PI}(t) &= K_P^{fp} e(t) + K_I^{fi} \int e(t)\mathrm{d}t, \\ u_{PD}(t) &= K_P^{fp} e(t) + K_D^{fd} \frac{\mathrm{d}e(t)}{\mathrm{d}t}, \end{aligned} \quad (6)$$

where K_P^{fp} is the proportional gain of fuzzy control modulation, K_I^{fi} is the integral gain of fuzzy control modulation, and K_D^{fd} is the derivative gain of fuzzy control modulation.

3.2. Design of FOFPID Controller. The FOFPID controller has evolved from fuzzy PID with noninteger order calculus. It consists of fractional calculus, fuzzy logic control, and two separate PI and PD controllers, and the structure of the FOFPID controller is shown in Figure 3. The structure only has two input variables, and fuzzy PI and fuzzy PD controllers share the same rule base to perform control actions. The controller has six parameters, including four scaling factors $\{K_u, K_r, K_i, K_d\}$ and two fractional orders $\{\mu, \lambda\}$. Deviation e and its variation of deviation $d^{\mu}/\mathrm{d}t^{\mu}$e are the two input parameters of fuzzy controller which are used to determine fuzzy rules. Two input scaling factors $\{K_u, K_r\}$ are the common input gains of FOFPID controller, modulating the input parameters to meet ranges of fuzzy membership functions, and another two scaling factors $\{K_i, K_d\}$ are fuzzy PI controller gain and fuzzy PD controller gain, respectively. Fractional calculus is applied into derivative and integral parts through fractional orders μ and λ, as the distinction of fuzzy FID and FOFPID. For the fuzzy PID controller, μ and λ are equal to 1, whereas μ and λ are fractional in the FOFPID controller.

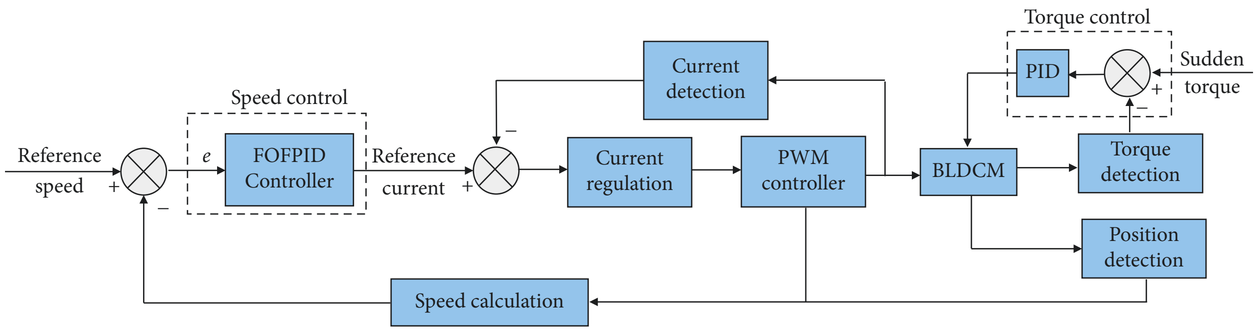


Figure 1: Modular modelling schematic of the BLDCM system.

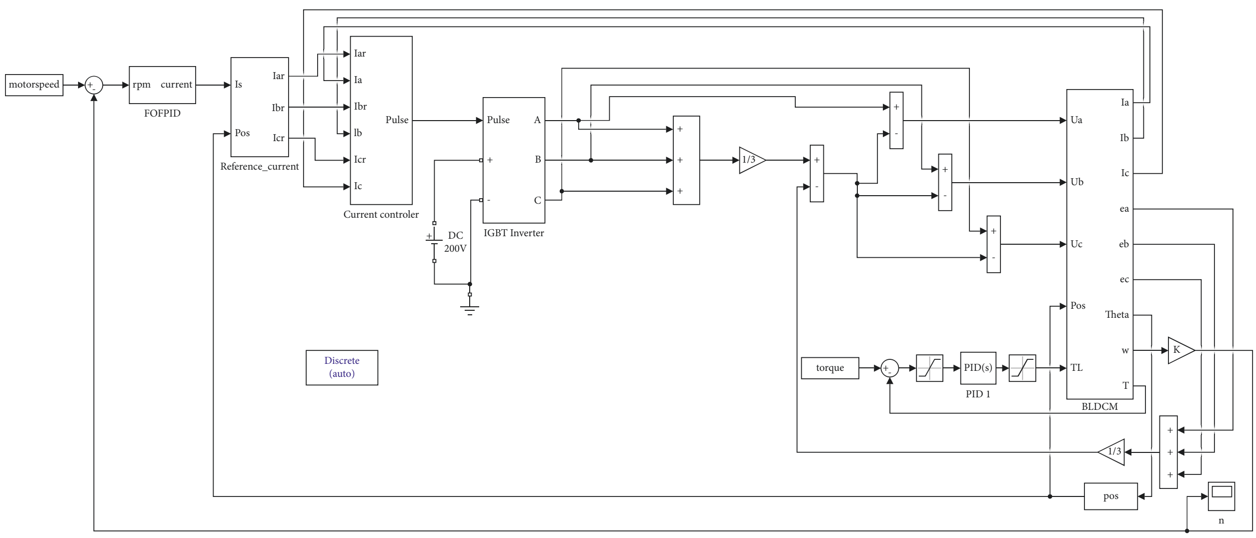


Figure 2: Simulation model of the BLDCM system in Simulink.

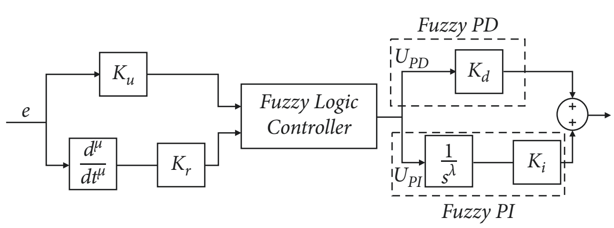


Figure 3: Block diagram of the FOFPID controller.

The FOFPID controller could be expressed as [23]

$$K_u e + K_r \frac{d^\mu}{dt^\mu} e = K_d u_{PD} + K_i \frac{1}{s^\lambda} u_{PI}, \quad (7)$$

where d^μ/dt^μ is the differentiator, $1/s^\lambda$ is the integrator, U_{PD} is the control action of the fuzzy PD controller, and U_{PI} refers to control actions of the fuzzy PI controller.

Formulating fuzzy control rules is the key of designing a fuzzy controller which includes selecting the linguistic variables of input and output parameters, defining the fuzzy subset of fuzzy variables, and establishing the control rules of fuzzy controller [31]. The rule table of the fuzzy controller is a set of fuzzy condition statement with several linguistic variables. Choosing appropriate linguistic variables to describe input and output parameters could make the establishment of control rules more accurate, and defining the fuzzy subset is to determine the shape of membership functions. Figure 4 shows the membership functions of input and output of fuzzy logic control. Symmetrical triangular membership functions were adopted for three inputs factors, including negative (N), zero (Z), and positive (P) variables, while singleton consequents were adopted for five output factors, including NB (negative big), NM (negative medium), ZO (zero), PM (positive medium), and PB (positive big) variables.

Based on the membership functions of inputs and outputs, the fuzzy control rules of fuzzy PI and fuzzy PD controllers are described as follows:

Rule 1: if K_ue is N and $K_r d^\mu/dt^\mu$e is N, then U_{PD}/U_{PI} is NB.

Rule 2: if K_ue is N and $K_r d^\mu/dt^\mu$e is Z, then U_{PD}/U_{PI} is NM.

Rule 3: if K_ue is N and $K_r d^\mu/dt^\mu$e is P, then U_{PD}/U_{PI} is ZO.

Rule 4: if K_ue is Z and $K_r d^\mu/dt^\mu$e is N, then U_{PD}/U_{PI} is NM.

Rule 5: if K_ue is Z and $K_r d^\mu/dt^\mu$e is Z, then U_{PD}/U_{PI} is ZO.

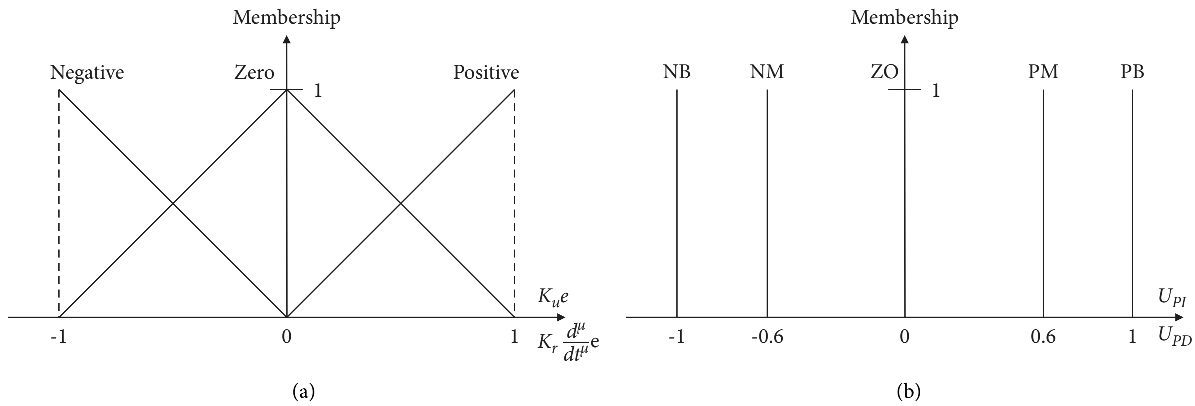


FIGURE 4: (a) Input membership functions. (b) Output membership functions.

TABLE 1: Six parameters of the FOFPID controller.

$\mathbf{K_u}$	15.172
$\mathbf{K_r}$	6.489
$\mathbf{K_i}$	12.239
$\mathbf{K_d}$	1.011
μ	0.348
λ	0.624

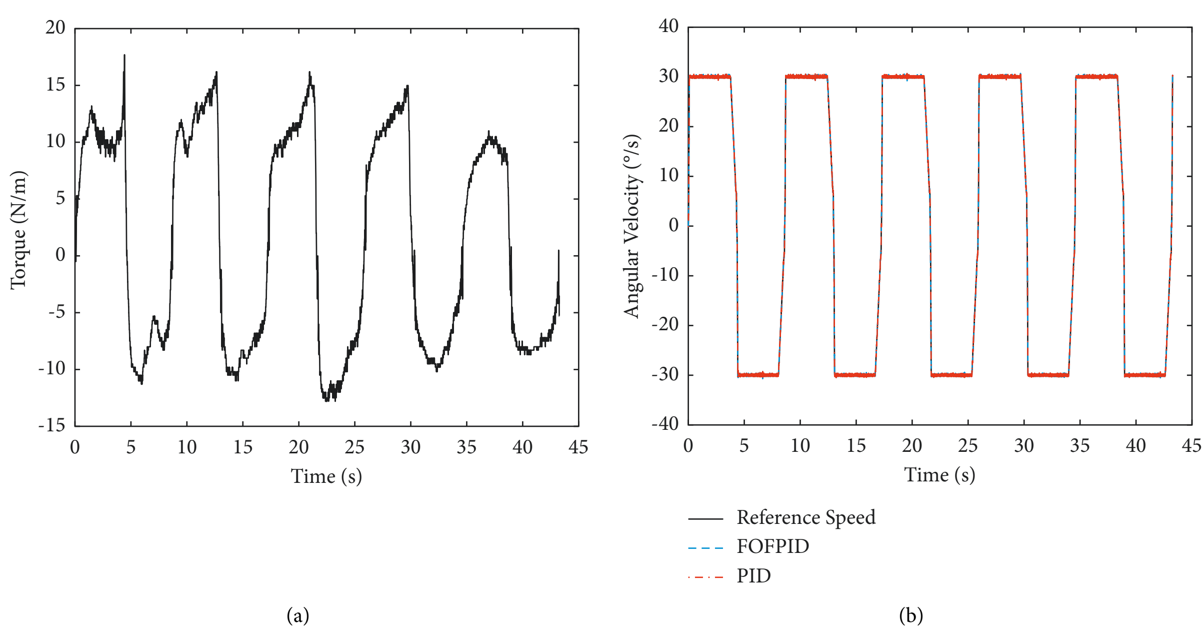
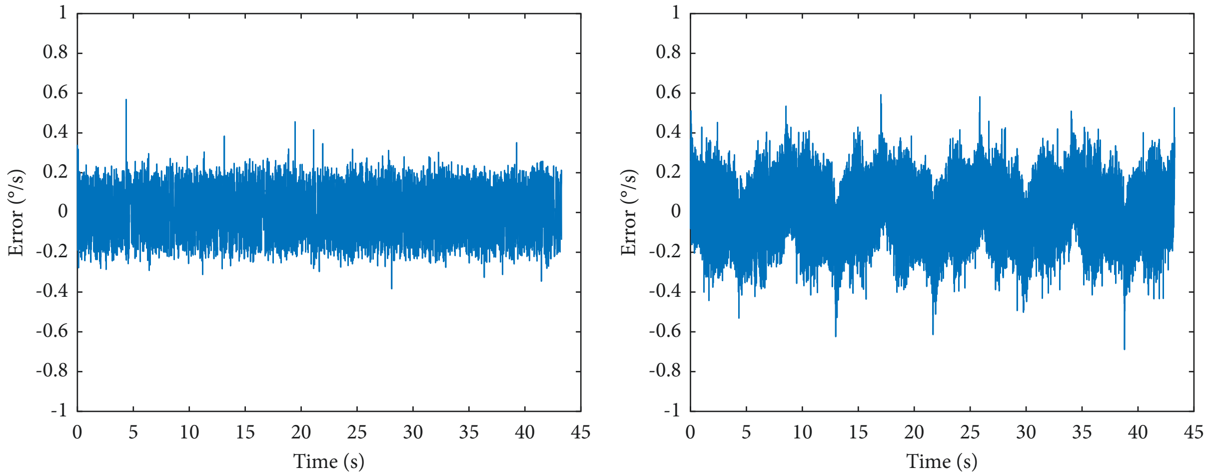
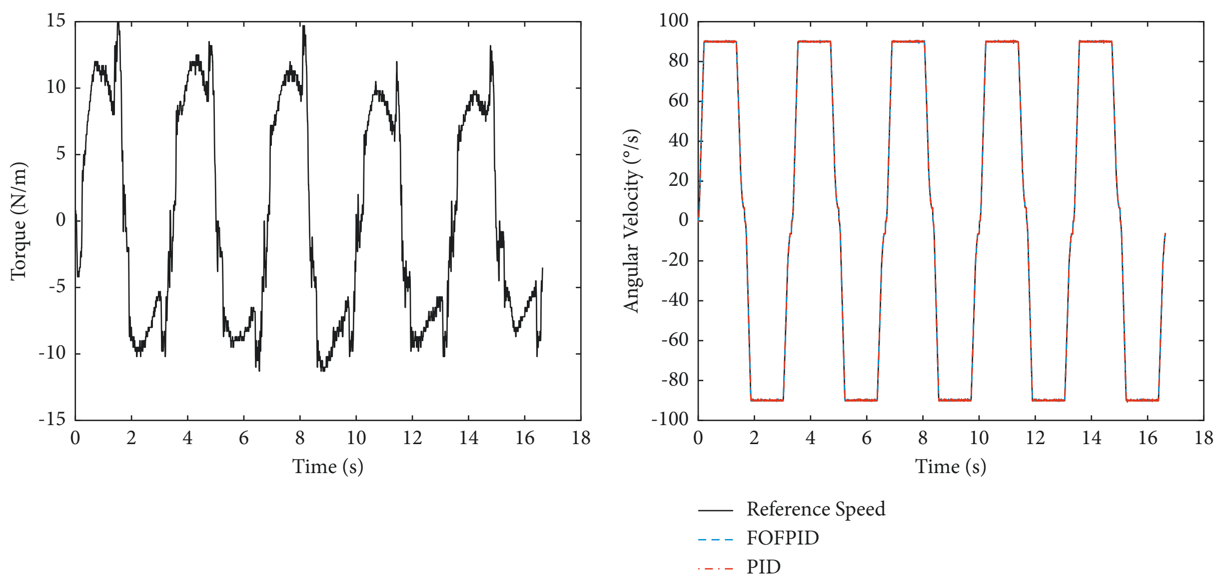


FIGURE 5: Continued.



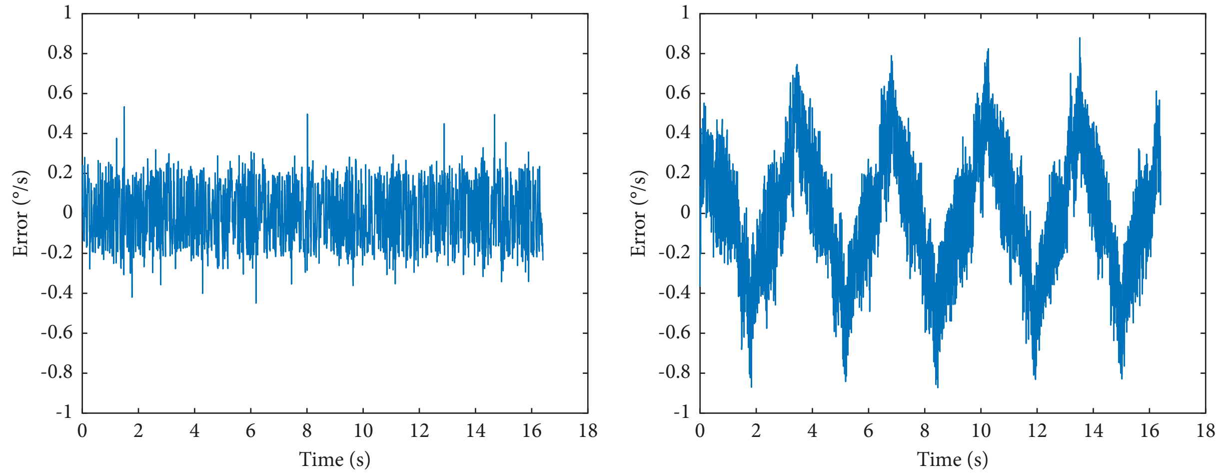
(c) (d)

Figure 5: Simulation results under 30°/s. (a) Elbow joint torque. (b) Comparison between reference speed, speed response of the FOFPID controller, and speed response of the PID controller. (c) Error between reference speed and speed response of the FOFPID controller. (d) Error between reference speed and speed response of the PID controller.



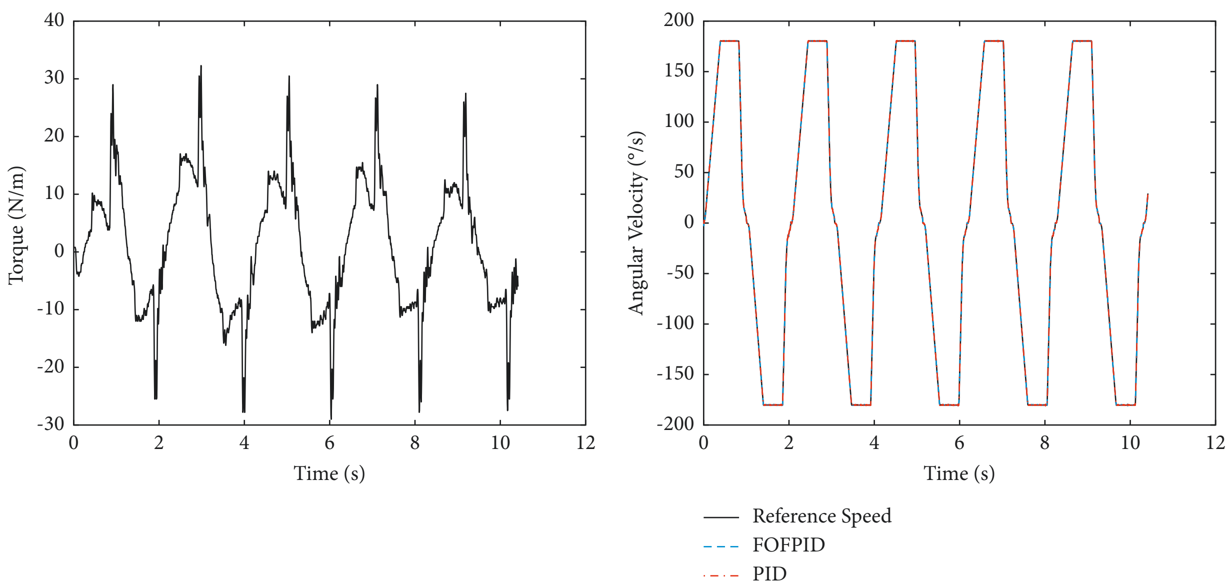
(a) (b)

Figure 6: Continued.



(c) (d)

Figure 6: Simulation results under 90°/s. (a) Elbow joint torque. (b) Comparison between reference speed, speed response of the FOFPID controller, and speed response of the PID controller. (c) Error between reference speed and speed response of the FOFPID controller. (d) Error between reference speed and speed response of the PID controller.



(a) (b)

Figure 7: Continued.

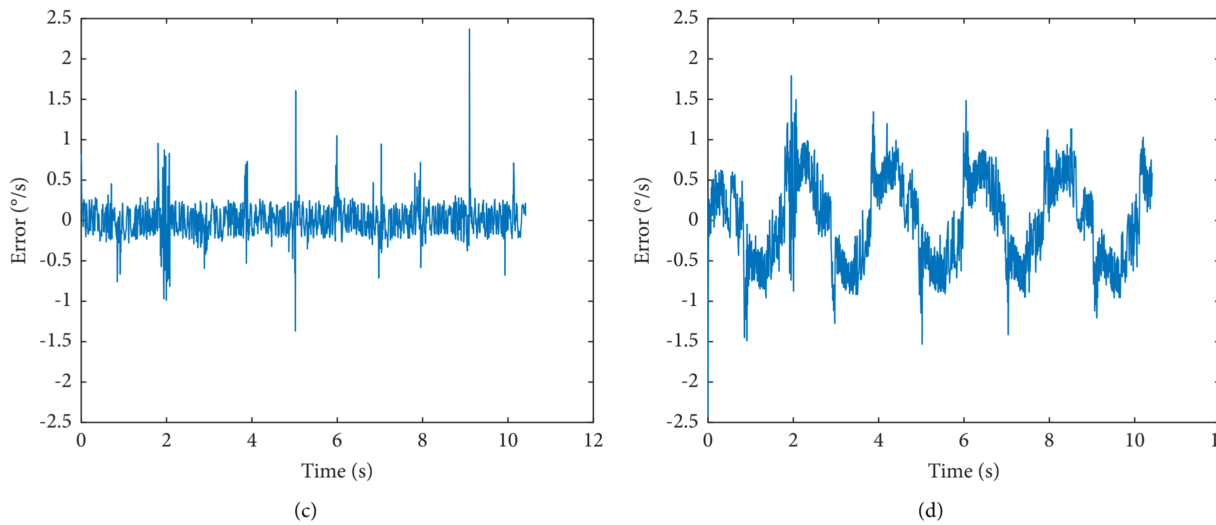


FIGURE 7: Simulation results under 180°/s. (a) Elbow joint torque. (b) Comparison between reference speed, speed response of the FOFPID controller, and speed response of the PID controller. (c) Error between reference speed and speed response of the FOFPID controller. (d) Error between reference speed and speed response of the PID controller.

Rule 6: if K_ue is Z and $K_r d^\mu/dt^\mu$e is P, then U_{PD}/U_{PI} is PM.

Rule 7: if K_ue is P and $K_r d^\mu/dt^\mu$e is N, then U_{PD}/U_{PI} is ZO.

Rule 8: if K_ue is P and $K_r d^\mu/dt^\mu$e is Z, then U_{PD}/U_{PI} is PM.

Rule 9: if K_ue is P and $K_r d^\mu/dt^\mu$e is P, then U_{PD}/U_{PI} is PB.

Six tuned gains of the FOFPID controller ($\{K_u, K_r, K_i, K_d\}$ and $\{\mu, \lambda\}$) which were adopted to control speed of the simulation model are listed in Table 1.

4. Simulation Results

In this section, the simulation results and result analysis would be presented to explore the stability of the BLDCM system using the FOFPID controller.

4.1. Reference Data Collection. Reference speed and load torque are the two input parameters which were collected from the Isomed2000 isokinetic device. The experiment is focused on the isokinetic movement of the elbow joint, and the elbow adapter is assembled on the fixed axis. The experimenter was seated on the seat with the right hand holding the handle of adapter and was required to perform maximum voluntary torque (MVT) during the elbow joint isokinetic movement. The movement is a kind of elbow joint flexion and extension movement. Eight sets of experimental data were collected under the different joint angular velocities of 30, 45, 60, 75, 90, 120, 150, and 180°/s, respectively, and the experimenter finished five groups of reciprocating motions for each preset speed. Each group includes an elbow flexion and extension process which could be considered as motor forward rotation and reverse rotation. Isokinetic movement is divided into three stages: acceleration stage, isokinetic stage, and deceleration stage. Under different reference speeds (joint angular velocities), the training effects are various and the stability is also affected differently by load torque. The fluctuations of velocities and joint torque were recorded by sensors of the isokinetic exercise device.

4.2. Simulation Results. For the simulation experiment in Simulink, the FOFPID controller was applied to control the reference speed of the BLDCM; meanwhile, a PID controller was used to control the load torque of the system. According to the simulation model in Figure 2, the control results under different reference speeds could be obtained separately. Figures 5–7 show the following under angular velocities of 30, 90, and 180°/s: (a) the elbow joint torque resisting the resistance movement; (b) the comparison between reference speed, speed response of the FOFPID controller, and speed response of the PID controller; (c) error between the reference speed and the FOFPID controller speed response; (d) error between the reference speed and the PID controller speed response. These three figures could represent the variation tendency of joint torque and speed response from low speed to high speed.

For isokinetic movement, the joint torque is related to the muscle force of the experimenter and the preset speed of the equipment. In the experiment, the load torque is the elbow joint torque under MVT. The torque values in Figures 5(a), 6(a), and 7(a) show irregular periodic fluctuations and have spikes between two isokinetic stages especially in the high-speed conditions. This is because the experimenter should follow the speed as quick as possible and switch the directions rapidly. Thus, the torque gained a peak value when the motor went from forward to reverse or from reverse to forward (acceleration stages and deceleration stages) where the accelerations are quite large when the angular velocities are high relatively. The load torque in isokinetic stages remains in a relatively stable state compared with other two stages. These load torque values are the input disturbances of the BLDCM system.

The comparison between different reference speeds, speed response of the FOFPID controller, and speed response of the PID controller provides an intuitive presentation of controller's tracking characteristics. Figures 5(b), 6(b), and 7(b) show a good tracking performance of the FOFPID controller and PID controller that all the speed responses under 30, 90, and 180°/s joint angular velocities could follow the reference speeds successfully. To compare the stability of the FOFPID controller and PID controller, the errors between reference speed and FOFPID controller speed response are described in Figures 5(c), 6(c), and 7(c), and the errors between reference speed and PID controller speed response are given in Figures 5(d), 6(d), and 7(d). These errors revealed the tracking ability during the simulation processes. It is observed that the errors of the FOFPID controller under three angular velocities have uniform error ranges around ±0.2°/s, while the errors of the PID controller have obvious fluctuations along with the changes of speed and torque. Also, comparing the errors of the FOFPID controller with the errors of the PID controller, the stability of the FOFPID controller is better than that of the PID controller for the smaller fluctuation ranges. In Figure 7(c), there exists some abnormal value which exceeds the error range mainly caused by the sudden peak values of load torque during acceleration stages and deceleration stages.

In order to reflect the performance of the FOFPID controller and PID controller more intuitively, data statistics and distribution results of errors are shown in Figures 8 and 9 which display the statistics and distribution results of eight speed errors. The small boxes for each velocity are the main data distribution regions which are limited between the upper interquartile and lower interquartile. Two segments above and under the boxes are upper adjacent and lower adjacent to intercept extreme outliers. The red plus signs which extend to the top and bottom directions are the mild outliers. From Figure 8, the distribution of errors under eight angular velocities remains between ±0.5°/s, while the error distribution in Figure 9 shows an increasing trend. This demonstrates that the FOFPID controller could maintain the stability of the BLDCM system regardless of the velocity change or torque disturbance; however, the stability of the PID controller would decrease as the preset velocity increases. Although some mild outliers exist in Figure 8, all the mild outliers were produced during acceleration stages and deceleration stages as shown in Figure 7(c), where the deviations tend to occur but have small impact on the system especially for isokinetic stages. Meanwhile, the upper adjacent, median, and lower adjacent values of each error box were counted as shown in Table 2. It is obvious to observe the error comparison and system stability between the FOFPID controller and PID controller.

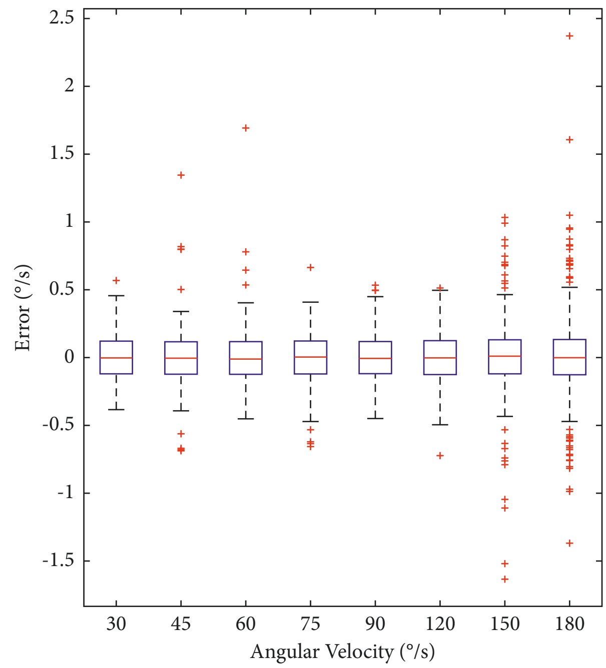


Figure 8: The statistics and distribution of errors between reference speed and FOFPID controller speed response under various joint angular velocities.

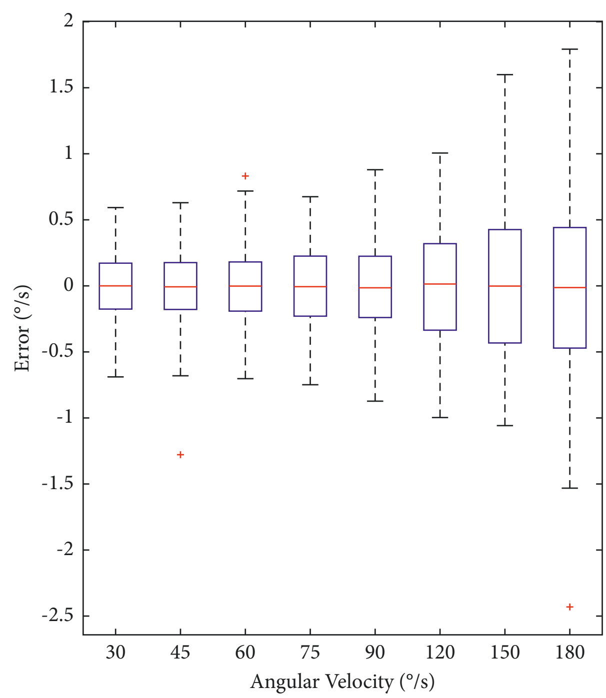


Figure 9: The statistics and distribution of errors between reference speed and PID controller speed response under various joint angular velocities.

Table 2: Statistics of error upper boundary, lower boundary, and median under eight speeds.

Speed (°/s)		30	45	60	75	90	120	150	180
Upper adjacent	FOFPID	0.456	0.340	0.404	0.409	0.449	0.496	0.464	0.518
	PID	0.593	0.630	0.718	0.675	0.879	1.006	1.599	1.792
Median	FOFPID	−2.6e − 03	−4.5e − 03	−1.1e − 02	4.0e − 03	−6.0e − 03	−2.3e − 03	1.0e − 02	−9.8e − 04
	PID	4.9e − 05	−7.2e − 03	−1.6e − 03	−6.2e − 03	−1.5e − 02	1.5e − 02	−1.5e − 03	−1.3e − 02
Lower adjacent	FOFPID	−0.384	−0.393	−0.452	−0.472	−0.450	−0.495	−0.434	−0.471
	PID	−0.689	−0.681	−0.703	−0.749	−0.873	−0.997	−1.058	−1.532

5. Conclusion

The simulation experiment in this paper is to explore the stability of isokinetic control of the BLDCM system and realize isokinetic control of the BLDCM system in MATLAB/Simulink environment. As a nonlinear system, the FOFPID controller was applied to implement the isokinetic control based on the BLDCM system. This study of isokinetic control discussed the stability performance of the BLDCM system under different preset speeds and resistance moments according to the comparison between the FOFPID controller and PID controller. The biggest challenge is to track the preset speed under continuous and various load torques. From the simulation results in Section 4, the FOFPID controller has better stability performance compared with the PID controller which could maintain the system errors around ±0.5°/s for all joint angular velocities.

According to the comparison between reference speed and FOFPID controller speed response, the proposed controller could obtain a better dynamic performance under various angular velocities and complex load torque. In the future, the proposed controller needs to be further optimized to enhance the stability and robustness by improving the parameters and membership functions of variables and further considering the characteristics of external disturbance. Meanwhile, the simulation results should be experimentally validated to verify the effectiveness of the proposed controller in real experiments.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

An earlier version of this paper has been presented as conference in 2019 IEEE 9th Annual International Conference on Cyber Technology in Automation Control and Intelligent Systems (CYBER).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (nos. 51575338, 51575407, 51475427, and 61733011) and the Fundamental Research Funds for the Central Universities (17 JCYB03).

References

[1] T. J. Suchomel, S. Nimphius, and M. H. Stone, "The importance of muscular strength in athletic performance," *Sports Medicine*, vol. 46, no. 10, pp. 1419–1449, 2016.

[2] M. B. Undheim, C. Cosgrave, E. King et al., "Isokinetic muscle strength and readiness to return to sport following anterior cruciate ligament reconstruction: is there an association? A systematic review and a protocol recommendation," *British Journal of Sports Medicine*, vol. 49, no. 20, pp. 1305–1310, 2015.

[3] L. Wang, Z. Zhang, Y. Yu, and G. Chen, "Study on the control of variable resistance for isokinetic muscle training system," *Technology and Health Care*, vol. 25, pp. 45–52, 2017.

[4] K. A. Taylor, J. D. Wiles, D. A. Coleman, P. Leeson, R. Sharma, and J. M. O'Driscoll, "Neurohumoral and ambulatory haemodynamic adaptations following isometric exercise training in unmedicated hypertensive patients," *Journal of Hypertension*, vol. 37, 2018.

[5] T. Bohman, M. Tegern, A. Halvarsson, L Broman, and H Larsson, "Concurrent validity of an isokinetic lift test used for admission to the Swedish Armed Forces," *PLoS ONE*, vol. 13, no. 11, Article ID e0207054, 2018.

[6] Z. Dvir, *Isokinetics: Muscle Testing, Interpretation and Clinical Applications*, Elsevier-Churchill Livingstone, Edinburgh, Scotland, 2nd edition, 2004.

[7] G. J. Davies, *A Compendium of Isokineties in Clinical Usage and Rehabilitation Techniques*, S Publisher, La Crosse, WI, USA, 4th edition, 1992.

[8] N. H. Kim, O. Yang, and M. H. Kim, "BLDC motor control algorithm for industrial applications using a general purpose processor," *J. Power Electron.*vol. 7, no. 2, pp. 132–139, 2007.

[9] M. Nerat and D. Vrancic, "A novel fast-filtering method for rotational speed of the BLDC motor drive applied to valve actuator," *IEEE*, vol. 21, no. 3, pp. 1479–1486, 2016.

[10] W. Lee, J. H. Kim, W. Choi, and B. Sarlioglu, "Torque ripple minimization control technique of high-speed single-phase brushless DC motor drive for electric turbocharger," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 99, 2018.

[11] T. S. Sivarani, S. Joseph Jawhar, C. Agees Kumar, and K. Prem Kumar, "Novel Bacterial Foraging-Based ANFIS for Speed Control of Matrix Converter-Fed Industrial BLDC Motors Operated under Low Speed and High torque," *Neural Computing and Applications*, vol. 29, 2016.

[12] K. J. Astrom and T. Haggulund, *PID Controllers: Theory, Design and Tuning*, International Society for Measurement and Control, North Carolina, NC, USA, 2nd edition, 1995.

[13] S. Thamizmani and S. Narasimman, "Design of fuzzy PID controller for brushless DC motor," *Int J Emerg Res Manage Technol*, vol. 9359, pp. 68–78, 2014.

[14] C. Xia, G. Jiang, W. Chen, and T. Shi, "Switching-gain adaptation current control for brushless DC motors," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2044–2052, 2016.
[15] Y. Mandel and G. Weiss, "Adaptive internal model-based suppression of torque ripple in brushless DC motor drives," *Systems Science & Control Engineering*, vol. 3, no. 1, pp. 162–176, 2015.
[16] X. Wang, T. Fu, and X. Wang, "BLDC motor control system based on quadratic single neuron adaptive PID algorithm," *International Journal of Advanced Pervasive and Ubiquitous Computing*, vol. 6, no. 1, pp. 12–25, 2017.
[17] Z. Xu, X. Zhou, H. Wu, X. Li, and S. Li, "Motion planning of manipulators for simultaneous obstacle avoidance and target tracking: an RNN approach with guaranteed performance," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 3887–3897, 2022.
[18] J. Dong, D. Li, and Y. Zhang, "Active disturbance rejection control for TITO systems," in *Proceedings of the 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, Xi'an, China, July 2020.
[19] Z. Xu, X. Li, S. Li, H. Wu, and X. Zhou, "Dynamic neural networks based adaptive optimal impedance control for redundant manipulators under physical constraints," *Neurocomputing*, vol. 471, no. 2, 2021.
[20] Q. Zhang, H. Guo, Y. Liu et al., "Robust plug-in repetitive control for speed smoothness of cascaded-PI PMSM drive," *Mechanical Systems and Signal Processing*, vol. 163, no. 8, Article ID 108090, 2022.
[21] X. Li, X. Zhou, Z. Xu, and G. Tang, "Chaos-based second-order bas for multimodal function optimization," *Filomat*, vol. 34, no. 15, pp. 5159–5172, 2020.
[22] R. H. Abiyev, O. Kaynak, T. Alshanableh, and F. Mamedov, "A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization," *Applied Soft Computing*, vol. 11, no. 1, pp. 1396–1406, 2011.
[23] K. S. Tang, K. F. Kim Fung Man, G. Guanrong Chen, and S. Kwong, "An optimal fuzzy PID controller," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 4, pp. 757–765, 2001.
[24] K. Kiguchi, T. Tanaka, and T. Fukuda, "Neuro-fuzzy control of a robotic exoskeleton with EMG signals," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 481–490, 2004.
[25] P. Muthukumar, P. Balasubramaniam, and K. Ratnavelu, "T-S fuzzy predictive control for fractional order dynamical systems and its applications," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 751–763, 2016.
[26] V. Kumar and K. P. S. Rana, "Nonlinear adaptive fractional order fuzzy PID control of a 2-link planar rigid manipulator with payload," *Journal of the Franklin Institute*, vol. 354, Article ID S0016003216304227, 2016.
[27] V. Kumar, K. P. S. Rana, J. Kumar, and P. Mishra, "Self-tuned robust fractional order fuzzy PD controller for uncertain and nonlinear active suspension system," *Neural Computing and Applications*, vol. 30, 2016.
[28] H. C. Lovatt, M. L. McClelland, and J. M. Stephenson, "Comparative Performance of singly salient reluctance, switched reluctance, and induction motors," in *Proceedings of the The Eighth International Conference on Electrical Machinesand Drives*, pp. 361–365, Cambridge, UK, September 1997.
[29] P. Crnosija, R. Krishnan, and T. Bjazic, "Transient performance based design optimization of PM brushless DC motor drive speed controller," in *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005*, vol. 3, pp. 881–886, Dubrovnik, Croatia, June 2005.
[30] P. Mishra, V. Kumar, and K. P. S. Rana, "A fractional order fuzzy PID controller for binary distillation column control," *Expert Systems with Applications*, vol. 42, no. 22, Article ID S0957417415004686, 2015.
[31] A. Katbab, "Fuzzy logic and controller design-a review," in *Proceedings of the Southeastcon'95: Visualize the Future*, pp. 443–449, Raleigh, NC, USA, March 1995.