International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 20s, 2025
https://theaspd.com/index.php

An Artificial Neural Network Solution To A Higher-Order Fractional Linear Integro-Differential Problem Using Deep Learning

L.K.Suresh Kumar[1], P.Vijayapal Reddy[2], Bolukonda Prashanth[3], Anuradha Nandula[4], P.Preeti Payal[5]
[1]Department of Computer Science and Engineering, University College of Engineering(A),Osmania University Hyderabad 500007,Telangana,India
[2]Department of Computer Science and Engineering, Keshav Memorial College of Engineering, Hyderabad Telangana,India
[3]Department of CSE(Data Science),Sumathi Reddy Institute of Technology for Women,Warangal,Telangana,India.
[4]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India.
[5]Department of MCA , Princeton PG College Of Information Technology,Hyderabad

Abstract: *To put it another way, the idea of organizing an attention was drawn to the best iterative first-order approach for estimating solutions to the origin fractional issue. Furthermore, a few computer simulation models show how accurate and useful the suggested iterative method is. When compared to traditional methods, the exceptional achieved numerical figures easily demonstrate the efficiency and skill of artificial neural network techniques. One new and exciting topic of study in the fields of machine learning and numerical analysis is the use of ANN to solve fractional higher-order linear integro-differential equations (IHODEs).. Fractional-order integro-differential equations are extensions of classical differential equations, where derivatives are of non-integer order, often incorporating memory and hereditary properties, which makes them useful in modeling complex systems in various fields, such as physics, engineering, and finance*

Keywords: *The fractional derivative of Caputo, cost function, learning technique, artificial neural network approach, and linear integro-differential equation of increasing order*

1 INTRODUCATION

Using the well-established basis of fractional calculus disputes, differentiation and integration were, as far as we know, extended to random non-integer (real/complex) order. Over the past 20 years, fractional calculus has been the focus of a lot of study in the field of modern mathematics. For this reason, in applied mathematics and other pertinent scientific and professional fields, a common problem is the fractional-order integro-differential equations (FOIDEs) problem. This is one of the main reasons why a wide range of scientists and academics have found the problem addressed in the present study to be more intriguing. It is well known that solving many beginning possible issues with border values it might be difficult to work openly with fractional derivatives. Fractional differential equations (FDEs) have become an important tool in modeling systems that exhibit memory and non-local behavior. Unlike classical differential equations, which involve integer-order derivatives, fractional derivatives generalize differentiation to non-integer orders, allowing for a more accurate representation of physical phenomena that depend on past states of the system. These equations are widely used in fields such as physics, engineering, biology, and finance to model processes involving diffusion, viscoelasticity, and anomalous diffusion, among others.

Among the many types of fractional differential equations, fractional higher-order linear integro-differential equations (IHODEs) are of particular interest due to their ability to describe systems where both past and future interactions contribute to the current state. An example of such an equation is:

$$\frac{d^\alpha y(t)}{dt^\alpha} + \int_0^t K(t-\tau)y(\tau)d\tau = f(t),$$

Where $\frac{d^\alpha y(t)}{dt^\alpha}$ represents a fractional derivative of order $\alpha, K(t-\tau)$ is a kernel function capturing the memory effects, and $f(t)$ is a forcing function. Solving such equations is generally more difficult than solving integer-order differential equations due to the involvement of non-local operators (the fractional derivative and the integral) and the complexity of their interactions.

Traditionally, numerical approaches like finite difference, finite element, or spectral methods are used to solve fractional IHODEs.. While these approaches are well-established, they often require discretization of the domain, which can introduce errors and computational inefficiencies, especially for complex or high-dimensional problems. Additionally, fractional derivatives are not as straightforward to discretize as integer-order derivatives, and thus these methods may struggle to achieve high accuracy.
Artificial Neural Networks (ANNs) have become a potent substitute for conventional numerical techniques in recent years. We provide an approach where a neural network is trained to reduce the inaccuracy between the equation's right and left sides, which contain integrals and fractional derivatives. The neural network learns a correct approximation of the answer in this way, which may have benefits for flexibility and computational efficiency.
The structure of this document is as follows: A synopsis of fractional calculus and the associated ideas of fractional derivatives and integrals are given in Section 2. The approach for employing ANNs to solve fractional IHODEs is presented in Section 3, which includes information on the network architecture, training procedure, and fractional derivative implementation.

1.1 LITERATURE SECTION

The solution of fractional higher-order linear integro-differential equations (IHODEs) has garnered significant attention in recent years, owing to the increasing recognition of their importance in modeling real-world phenomena with memory and non-local interactions. Researchers have explored various analytical and numerical methods to solve such equations, with the development of ANN emerging as a promising alternative to traditional techniques. This section provides an overview of the key contributions in the field, focusing on the use of fractional calculus, traditional solution methods, and the recent advancements in applying ANNs for solving fractional IHODEs.
1. Fractional Calculus and Fractional Differential Equations
The IHODEs, are often used in areas such as:
1. Viscoelasticity: Modeling the stress-strain relationship in materials with memory (e.g., polymers, biological tissues).
2. Anomalous Diffusion: Describing diffusion processes where the classical laws of diffusion do not hold, such as in porous media or complex systems.
3. Biological Systems: Modeling processes such as signal propagation in neurons or the spread of diseases, where memory effects and non-local interactions are essential.
The general challenge in solving fractional IHODEs arises from the difficulty of discretizing fractional derivatives and integrals in a computationally efficient manner.
2. Traditional Numerical Methods for Fractional IHODEs
Traditional numerical methods for solving fractional IHODEs typically involve discretization techniques such as finite difference, finite element methods, and spectral methods. These approaches are extensions of their classical counterparts but adapted to handle fractional derivatives. Key methods include:
1. Finite Difference Methods (FDM): These methods discretize the domain and approximate derivatives using finite differences. Various techniques, such as the Grünwald-Letnikov and Caputo discretization methods, are commonly used to approximate fractional derivatives. While these methods are effective for simple problems, they struggle with high-order equations and non-local interactions in fractional IHODEs.
2. Finite Element Methods (FEM): FEM is widely used for solving partial differential equations, and its extension to fractional problems involves discretizing the spatial domain and using basis functions to approximate solutions. However, fractional derivatives complicate the assembly of the stiffness matrix and require sophisticated numerical techniques for accurate results.
3. Spectral Methods: These methods use a global representation of the solution (often through Fourier series or Chebyshev polynomials) to approximate the equation's solution. Spectral methods are highly accurate for smooth problems but may face difficulties with irregular domains or highly oscillatory solutions in fractional IHODEs.
Despite the successes of these traditional methods, they require careful discretization and can become computationally expensive, especially for complex, high-dimensional systems. Furthermore, accurately approximating fractional derivatives remains a key challenge in many numerical approaches.
3. Artificial Neural Networks for Solving Fractional Differential Equations

ANN have emerged as a novel approach to solving fractional differential equations, including IHODEs. ANNs have the advantage of learning the underlying dynamics of the system directly from data, without the need for discretization of the domain. Several studies have explored this approach for solving various types of fractional differential equations.

Key contributions in the application of ANNs for solving fractional differential equations include:

1. Raissi et al. (2019): They formulated the problem as a variational problem, where the network learns the solution by minimizing a loss function corresponding to the equation's residual. This method was applied to several fractional PDEs and demonstrated the power of ANNs in solving complex differential equations, including those with fractional derivatives.
2. Karniadakis et al. (2021: The PINN framework incorporates the differential equation directly into the loss function, enabling the network to learn the solution that satisfies both the data and the equation. This method has been successfully applied to various fractional and integer-order equations and is highly effective in solving problems without requiring large training datasets.
3. Jagtap and Karniadakis (2020): They extended the PINN framework to solve fractional PDEs by incorporating fractional derivatives directly into the network's loss function. This framework avoids the need for discretization of the fractional derivative and allows for accurate solutions to complex, high-order fractional differential equations.
4. Wang et al. (2023):. By using the CNN's ability to capture hierarchical patterns in data, they showed that ANNs can effectively model the intricate behaviors of systems described by fractional IHODEs.

4. Challenges and Opportunities

While ANNs have shown great promise in solving fractional IHODEs, several challenges remain:

1. Training Time: Deep neural networks, especially for complex fractional IHODEs, require significant computational resources and time for training. Interpretability: Unlike traditional numerical methods, the solutions obtained from neural networks can be difficult to interpret. Understanding how the network learns the underlying dynamics of the system and ensuring that the results are physically meaningful is an ongoing area of research.
2. Generalization: The ability of ANNs to generalize across different types of fractional equations and domains is still being investigated. Ensuring that trained networks can adapt to different kernel functions or fractional orders without retraining is a key area of interest.

The study and solution of fractional higher-order linear integro-differential equations (IHODEs) have gained considerable attention in recent years, especially as such equations provide a more accurate modeling framework for systems with memory, non-local interactions, and complex dynamics. These equations are widely used in physics, engineering, biology, and finance, but their solution remains a challenging task due to the involvement of fractional derivatives and integral terms. To address these challenges, both traditional numerical methods and modern techniques, such as Artificial Neural Networks (ANNs), have been explored for solving fractional IHODEs.

Despite the promising results, there are still several challenges in using ANNs to solve fractional IHODEs:

1. Training Time: Training deep neural networks, especially for large-scale problems, can be computationally expensive. Optimization methods such as gradient descent can be slow to converge, particularly for complex equations with many variables.
2. Interpretability: Neural networks, while effective at approximating solutions, often lack transparency in terms of understanding how the model arrives at the solution. This makes it challenging to interpret the results or validate them against physical intuition.

Generalization: The ability of ANNs to generalize across different types of fractional equations and parameter spaces is still under investigation. Ensuring that trained networks can be applied to a variety of problems without the need for retraining is an important direction for future research

2 INITIAL STEPS:

As previously stated, this study's main objective is to estimate the solution of the FOIDE issue using the ANN technique. This part provides a detailed explanation of the features in the theory of the ANN method, fractional calculus, and some necessary mathematical interpretations.

2.1 Calculus of fractions

A quick review of the literature indicates that the question "how can a function's derivative and integral be generalized to a non-integer order?" served as the impetus for the development of fractional calculus. Following this vague dispute, several academics focused on the mathematical expression for a derivative or integral of non-integer-order for a certain duration. Lastly, the first study on fractional-order derivatives was presented by Lacroix [20]. In the years that followed, a large number of scholars investigated fractional calculus and provided a wide range of useful explanations for integrals or non-integer order derivatives. The two definitions that appear to be most frequently used are the Caputo and Riemann-Liouville definitions. Since every derivative has a suitable operating range, it is unnecessary to identify which one has been used more frequently. Fractional order problems of initial value are better described by the Caputo definition [27]. We chose to employ Because the beginning circumstances were congruent, this investigation used Caputo's fractional definition. The Italian mathematician Caputo developed the following is a detailed definition of the Caputo fractional differential operator [4].

Definition 1: Assume that u(x) is a function that is continuously differentiable up to order k on the finite interval [a, b]. The fractional integral operator's definitions $I_{a,x}^{\alpha}$ and the Caputo derivative D_x^{α} of order $\alpha > 0$ are as follows:

$$ {}_aD_x^{\alpha}\, u(x) = \begin{cases} \dfrac{d^k u(x)}{dx^k}, \alpha = k \in N, \\ \dfrac{1}{\Gamma(k-\alpha)} \displaystyle\int_a^x \dfrac{u^{(k)}(\tau)}{(x-\tau)^{\alpha-k+1}}\, d\tau, x > a, 0 \le k-1 < \alpha < k, \end{cases} \quad (1) $$

$$ I_{a,x}^{\alpha} u(x) = \frac{1}{\Gamma(\alpha)} \int_a^x \frac{u(\tau)}{(x-\tau)^{1-\alpha}}\, d\tau, \quad (2) $$

respectively. The characteristics and functionality of the Caputo fractional operator have been the subject of several investigations. Here, we'll concentrate on its primary attributes and applications. It should be mentioned that the following characteristics are true and hence for all orders, the constant function's derivative is zero:

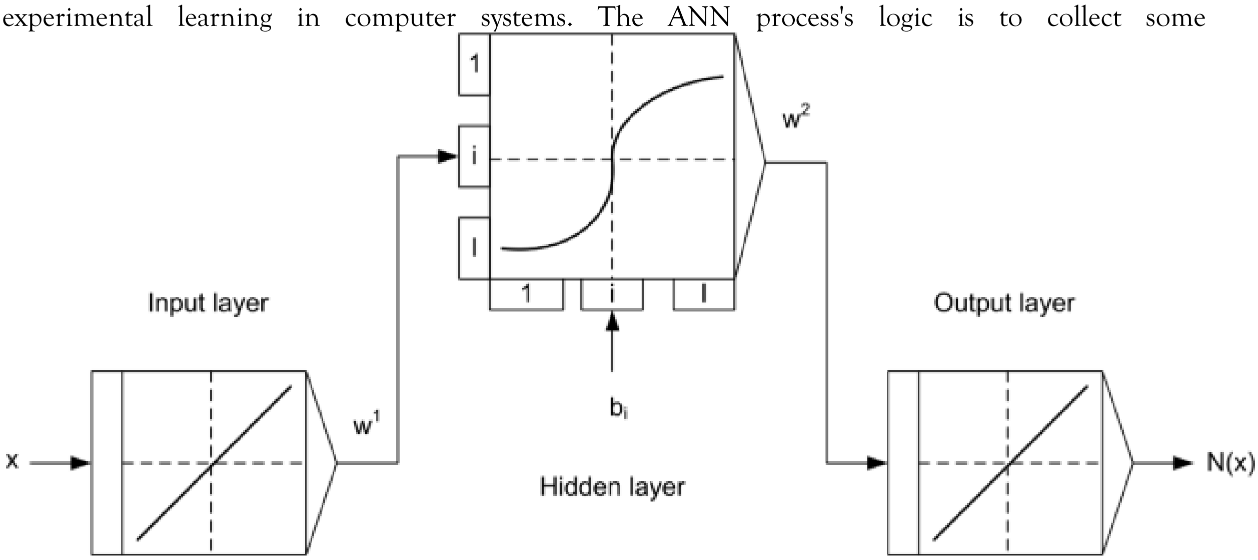
$$ {}_aD_x^{\alpha}\, [x^k] = \begin{cases} 0, \; k \in Z^+, k < [\alpha], \\ \dfrac{\Gamma(k+1)}{\Gamma(k+1-\alpha)} x^{k-\alpha}, x > ak \in Z^+, k \ge [\alpha], \end{cases} \quad (3) $$

$$ I_{0,x}^{\alpha}[t^k] = \frac{\Gamma(k+1)}{\Gamma(k+1-\alpha)} x^{k+\alpha}, k \in Z^+. \quad (4) $$

The notation $[\alpha]$ denotes the lowest integer that is bigger than or equal to the constant α in the aforementioned relations.

2.2 ANNs' fundamental structure

New methods are found throughout time, while others become less popular. An advanced, brain-inspired technology is called an artificial neural network (ANN) computing architecture designed to enable experimental learning in computer systems. The ANN process's logic is to collect some



An intended ANN architecture is shown in Figure 1.

training figures, after which an autonomous system is created that can get knowledge from the training set. From this vantage point, we shall investigate the "perceptron" as the brain architecture. Frank Rosenblatt, a researcher who built a noteworthy study program, made the proposal. In this kind of neural network, an input neurones get varying quantities of impulses. The input layer, the initial layer of the network, does not alter the values of the incoming signals. Using a set of network weights and biases, the neurones in the second layer also referred to as the hidden layer integrate their inputs. After that, they use an appropriate activation function to go through the concealed layer's nodes. Here, the oscillation of

the output of the hidden neurones is controlled by the sigmoidal activation function. The output of the network is then produced by sending the output from each node in the hidden layer to the final layer of neurones. Be aware that the identity function is used in both the input and output levels of this model, as it is in many others. A thorough explanation of the suggested methodology may be found in [6, 8]. One may understand the utility and inventiveness of ANNs by closely examining the brain architecture depicted in Fig. 1.
The following is a formulation of the description that is expounded upon in this section:

1. Unit of input layer:

$o_1^1 = x;$ (5)

2. hidden layer units:

$o_i^2 = f(net_i), i = 1, \dots, I,$ (6)

$net_i = x.w_i^1 + b_i,$

where the symbol f represents the sigmoid function;

3. Unit of the output layer:

$Net(x) = \sum_{i=1}^{I}(w_i^2 . o_i^2) = \sum_{i=1}^{I}(w_i^2 . f) w_i^1 . x + b_i))$. (7)

At this time, it is important to note that this pro-type network model can be effectively used to simulate the primary issue with a few minor adjustments.

3 Description of the proposed method

Because the FOIDEs may be used in a wide range of scientific fields, they have attracted a lot of attention. Finding precise answers to fractional issues is typically not an easy task. As a result, scientists are forced to make deductions using the proposed arbitrary numerical techniques. This section's main goal is to approximate the ANN-based answer to a certain class (see Fig. 1) of fractional integro-differential equations

$P(x).\,{}_aD_x^{\alpha_1}[u(x)] + Q(x).\,I_{a,x}^{\alpha_2}[u(t)] = H(x), 1 < \alpha_1, \alpha_2 \leq 2, a \leq x \leq b,$ (8)

. The method of power series (PS) predicts the solution to a minimisation (or maximisation) the problem at a particular location. The unknown function u(x) should now be rewritten in a suitable trial solution form before being approximated using the PS approach. This implies that the beginning conditions must be applied to employ this strategy, you must first address the origin problem. The following is the expression for the trial solution to the aforementioned equation:

$\tilde{u}(x) = \beta_1 + \beta_2 x + x^2 \sum_{i=1}^{I}(w_i^2 . f) w_i^1 . x + b_i))$. (9)

As the study progresses, an effort will be made to use the most appropriate BP machine learning approach to estimate the network parameter vectors b, w_1 and w_2.

3.1 Creating an issue involving minimization

As seen in the previous section, the trial solution (9), when combined with the proposed ANN architecture, can fully model and reproduce the fractional problem (8). It should be noted that before the neural network can be considered a potential solution the unknown function u(x) requires full training. Specifically, the learning goal is to find the right numerical values for the network's parameters such that the solution function may be approximated with high accuracy, namely w_i^1, w_i^2, and b_i(for i=1,..., n). Therefore, discretising the particular domain = (a, b) reduces the origin issue (8) to a similar reduction problem. Using the nodal points $xr = a + \frac{r(b-a)}{R}$ (for $r = 0, \dots, R, R \in N$), r represents a partition of the domain in this discretization process. The study is carried out under the assumption that $(a, b) = (0,1)$ for simplicity's sake. More broadly, the linearization operator $\frac{x}{b-a} + \frac{a}{a-b}, x \in (a, b)$ may completely convert any instance to this situation. The following appropriate form results from substituting the aforementioned trial answer an equation (8):

$P(x).\,{}_0D_x^{\alpha_1}[\beta_1 + \beta_2 x + x^2 \sum_{i=1}^{I}(w_i^2 . f) w_i^1 . x + b_i))]$ (10)
$+ Q(x).\,I_{0,x}^{\alpha_2}[\beta_1 + \beta_2 x + x^2 \sum_{i=1}^{I}(w_i^2 . f) w_i^1 . x + b_i))] = H(x), x \in \Omega$.

Here, the operators $D_x^{\alpha_1}$ and $I_{0,x}^{\alpha_2}$ must be dispersed over the series includes mathematically speaking, the non-linear activation function f it is quite challenging to compute the non-linear function f's fractional-order integral and derivative. We need to come up with a different plan to help us communicate the problem. For the calculation of currently, a recurrence relation is proposed for higher-order fractional derivatives [16]:

$f^{(n)}(x) = \sum_{k=1}^{n+1}(-1)^{k-1}\xi_k^n f^k$, (11)

$\xi_k^n = (k-1)\xi_{k-1}^{n-1} + k\xi_k^{n-1}$,

$\xi_k^n = 0, n < 0, k < 1, k > n+1.$

Table 1 displays given the initial values of n and k, the constant coefficients ξ_k^n. Equation (10) is substituted for an equation (11) and the result is simplified to provide the following outcome:

$$P(x).\,0^{D_x^{\alpha_1}}\left[x^2 \sum_{i=1}^{I}(w_i^2\ .\sum_{n=0}^{\infty}\frac{f^{(n)}(0)}{n!}(w_i^1.x+b_i)^n\right]$$
$$+ Q(x).\,I_{0,x}^{\alpha_2}\left[t^2 \sum_{i=1}^{I}(w_i^2\ .\sum_{n=0}^{\infty}\frac{f^{(n)}(0)}{n!}(w_i^1.x+b_i)^n\right]$$
$$+ Q(x).\frac{\beta_1}{\Gamma(1+\alpha_2)}.x^{\alpha_2} + Q(x).\frac{\beta_2}{\Gamma(2+\alpha_2)}.x^{1+\alpha_2} = H(x), x \in \Omega\,. (12)$$

Now, we have

$$P(x).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x^{j-\alpha_1+2}.(b_i)^{n-j}$$
$$+ Q(x).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_2+3)}.x^{j-\alpha_2+2}.(b_i)^{n-j}$$
$$+ Q(x).\frac{\beta_1}{\Gamma(1+\alpha_2)}.x^{\alpha_2} + Q(x).\frac{\beta_2}{\Gamma(2+\alpha_2)}.x^{\alpha_2+1} = H(x), x \in \Omega\,. (13)$$

The discovered positions $x_r(\text{for } r = 0,\ldots,R)$ are entered into an equation (13), completing this technique. Ultimately, the method known as the purpose of using the differentiable LMS is to improve The constant coefficients are shown in Table 1. ξ_k^n

n	k = 1	k = 2	k = 3	k = 4	k = 5
1	2	1	1	1	1
2	2	2	1	1	1
3	2	3	3	1	1
4	2	8	13	7	1
5	2	16	51	61	25

The optimisation technique is as follows:

$$E_r = \frac{1}{2}(P(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x_r^{j-\alpha_1+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_2+3)}.x_r^{j-\alpha_2+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\frac{\beta_1}{\Gamma(1+\alpha_2)}.x_r^{\alpha_2} + Q(x).\frac{\beta_2}{\Gamma(2+\alpha_2)}.x_r^{\alpha_2+1} - H(x_r))^2, x \in \Omega\,. \quad (14)$$

Our goal in the next section is to enhance this system by employing an appropriate mistake correction technique. Refer to reference [8] for further information.

3.1.1 The Suggested Method for Machine Learning

The well-known LMS rule was developed to transform the given integro-differential fractional beginning value issue in equation (8) into an optimisation model, as was rationally described above. The network error must be greatly optimised in the condensed network parameter space (weights and biases) in order to determine the created system's solution. The quadratic error function, which is the sum of the squared network errors, is minimised using the traditional gradient descent-based (BP) method. For $E_r(r = 0,\ldots,R,)$. Based on training a neural network by altering weights and biases, BP is a popular recurrent learning method. Initially, the network's parameters w_i^1, w_i^2, and b_i are training random constants with actual values. The differentiable function is then improved using the proposed supervised BP learning rule. $E = \sum_{r=0}^{R}$. In order to do this, the following procedure is defined for parameter w_i^2:

$w_i^2(\tau+1) = w_i^2(\tau) + \Delta w_i^2(\tau)$, (15)

$\Delta w_i^2(\tau) = -\eta.\frac{\partial E}{\partial w_i^2} + \gamma.\Delta w_i^2(\tau - 1), i = 1, \dots, I,$
$\frac{\partial E}{\partial w_i^2} = \sum_{r=0}^{R} \frac{\partial E_r}{\partial w_i^2},$
where τ, η, and γ stand for the momentum term, the number of repetition steps, and the learning rate, in that order. Additionally, the indices show each training subscript i label's updated and current weight parameters. $w_i^2(\tau + 1)$ and $w_i^2(\tau)$, in turn. The partial derivative $\frac{\partial E_r}{\partial w_i^2}$ is provided as follows to finish the learning process:

$$\frac{\partial E_r}{\partial w_i^2} = (P(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x_r^{j-\alpha_1+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_2+3)}.x_r^{j-\alpha_2+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\frac{\beta_1}{\Gamma(1+\alpha_2)}.x_r^{\alpha_2} + Q(x).\frac{\beta_2}{\Gamma(2+\alpha_2)}.x_r^{\alpha_2+1} - H(x_r))$$
$$\times (\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^1.(w_i^1)^j(b_i)^{n-j}$$
$$\left(P(x_r).\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x_r^{j-\alpha_1+2} + Q(x_r).\frac{\Gamma(j+3)}{\Gamma(j+\alpha_2+3)}.x_r^{j-\alpha_2+2}\right).$$

This modifying procedure is repeated for parameter w_i^1 in a manner akin to that for the weight parameter w_i^2 as follows:
$w_i^1(\tau + 1) = w_i^1(\tau) + \Delta w_i^1(\tau), (16)$
$\Delta w_i^1(\tau) = -\eta.\frac{\partial E}{\partial w_i^1} + \gamma.\Delta w_i^1(\tau - 1),$
Where

$$\frac{\partial E_r}{\partial w_i^1} = \sum_{r=0}^{R}((P(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x_r^{j-\alpha_1+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_2+3)}.x_r^{j-\alpha_2+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\frac{\beta_1}{\Gamma(1+\alpha_2)}.x_r^{\alpha_2} + Q(x).\frac{\beta_2}{\Gamma(2+\alpha_2)}.x_r^{\alpha_2+1} - H(x_r))$$
$$\times (\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.j.(w_i^1)^{j-1}(b_i)^{n-j}$$
$$\left(P(x_r).\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x_r^{j-\alpha_1+2} + Q(x_r).\frac{\Gamma(j+3)}{\Gamma(j+\alpha_2+3)}.x_r^{j-\alpha_2+2}\right))).$$

The bias parameter adjustment relations in this instance are the same as those in the previously provided example. Thus, we benefit
$b_i(\tau + 1) = b_i(\tau) + \Delta b_i(\tau), (17)$
$\Delta b_i(\tau) = -\eta.\frac{\partial E}{\partial b_i} + \gamma.\Delta b_i(\tau - 1),$
Where

$$\frac{\partial E_r}{\partial b_i} = \sum_{r=0}^{R}((P(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x_r^{j-\alpha_1+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\sum_{i=1}^{I}\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^j.\frac{\Gamma(j+3)}{\Gamma(j-\alpha_2+3)}.x_r^{j-\alpha_2+2}.(b_i)^{n-j}$$
$$+ Q(x_r).\frac{\beta_1}{\Gamma(1+\alpha_2)}.x_r^{\alpha_2} + Q(x).\frac{\beta_2}{\Gamma(2+\alpha_2)}.x_r^{\alpha_2+1} - H(x_r))$$
$$\times (\sum_{n=0}^{\infty}\sum_{j=0}^{n} C_n.\binom{n}{j} w_i^2.(w_i^1)^{j-1}.(n-j)(b_i)^{n-j-1}$$
$$\left(P(x_r).\frac{\Gamma(j+3)}{\Gamma(j-\alpha_1+3)}.x_r^{j-\alpha_1+2} + Q(x_r).\frac{\Gamma(j+3)}{\Gamma(j+\alpha_2+3)}.x_r^{j-\alpha_2+2}\right))).$$

The Volterra type is often analysed using the following technique:

$$P(x).{}_aD_x^{\alpha_1}[u(x)] + Q(x).I_{a,x}^{\alpha_2}[u(t)] = H(x), \quad a \le x \le b, \qquad (18)$$

Type equation here.The following is the selected trial solution for this issue that has been investigated:
$\tilde{u}(x) = \sum_{i=0}^{m} \beta_{i+1}x^i + x^{m+1}N(x). (19)$
An Equation (18) has been used in place of the trial solution (19) in order to follow the process, and other simplifications have been made. Consequently, when $x = x_r$, the parallel optimization system is satisfied. As previously mentioned, the BP learning rule may assist to mitigate the resulting issue. Please note that the relevant updating relations are not rebuilt here in order to avoid overstatement.

4 Illustrative examples

This section addresses two test sample problems to demonstrate the effectiveness and applicability of the recommended approach. To improve comprehension and demonstrate the correctness of the recommended strategy, a comparison with a technique described in [18] is done using the data collected below. The analytical program Matlab-R2013b has been used to perform the following mathematical computations. The following parameters were established:

1. η = 0.05 for the learning rate
2. γ = 0.01 is the momentum constant;
3. PS limitation N = 6.
4. R = 11 is the nodal point count.

As an illustration, 4.1 First, make the following assumptions about the Volterra-type higher-order fractional an integro-differential equation that is linear:

$$D_x^{1.5}u(x) + I_{0,x}^{2}u(t) = \frac{\Gamma(3)}{\Gamma\left(\frac{1}{5}\right)}x^{\frac{1}{2}} + \frac{\Gamma(3)}{\Gamma(5)}x^4, \ 0 \leq x \leq 1,$$

using the precise solution $u(x) = x^2 + 1$ and the basic criteria $u(0) = 1$, $u'(0) = 0$. Determining the network parameters w_i^1, w_i^2, and b_i (for $i = 1,\ldots,5$) using real-valued random constants would be essential to moving on. Following the application of the obtained updated data, net parameters are successively modified for τ = 1000. Obtained results are displayed in Table 2, which shows how accurate the study's approach was.

Fig. 2 shows a visualization of the shown total network error E. Furthermore, Fig. 3 plots the proximal and accurate answers for a range of repetition counts. An absolute differences between the precise and approximate answers are shown in Fig. 4. $\mathbf{E_{mid}}$ function in Figure 5 shows how well the planned ANN structure performs for a variety of control bases.

Table 2: Example 4.1's numerical results (for i=2)

x	The answer		Absolute mistake
	Exact	Approximate	
0.1	1.0100	1.010031652495624	3.1652×10^{-5}
0.2	1.0400	1.040050245132188	5.0245×10^{-5}
0.3	1.0900	1.090051494824863	5.1494×10^{-5}
0.4	1.1600	1.160037912535227	3.7912×10^{-5}
0.5	1.2500	1.250013755435422	1.3755×10^{-5}
0.6	1.3600	1.360009123128541	9.1231×10^{-6}
0.7	1.4900	1.490028884324511	2.8884×10^{-5}
0.8	1.6400	1.640038231158423	3.8231×10^{-5}
0.9	1.8100	1.810030512245391	3.0512×10^{-5}

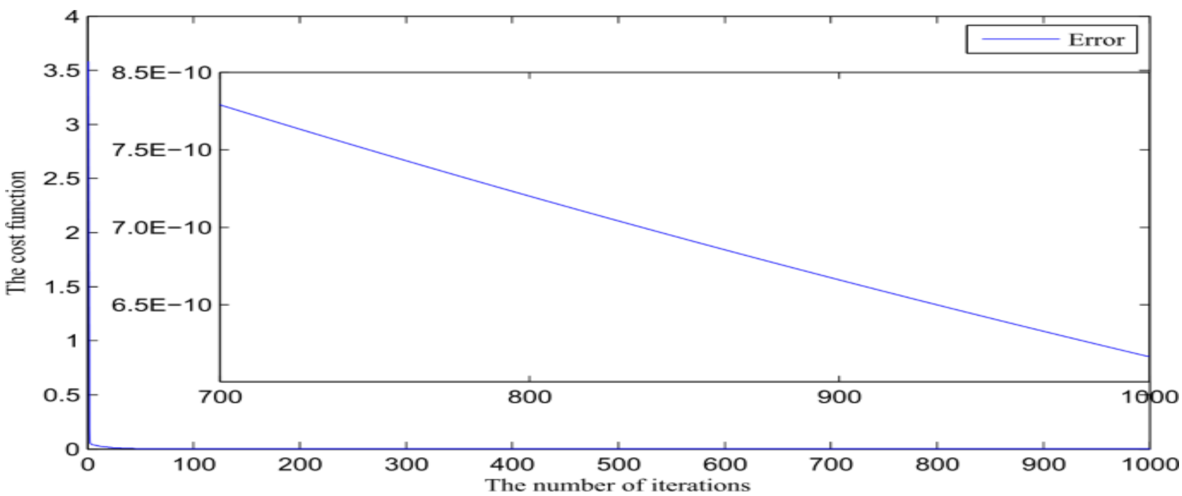


Figure 2: Example 4.1's error function

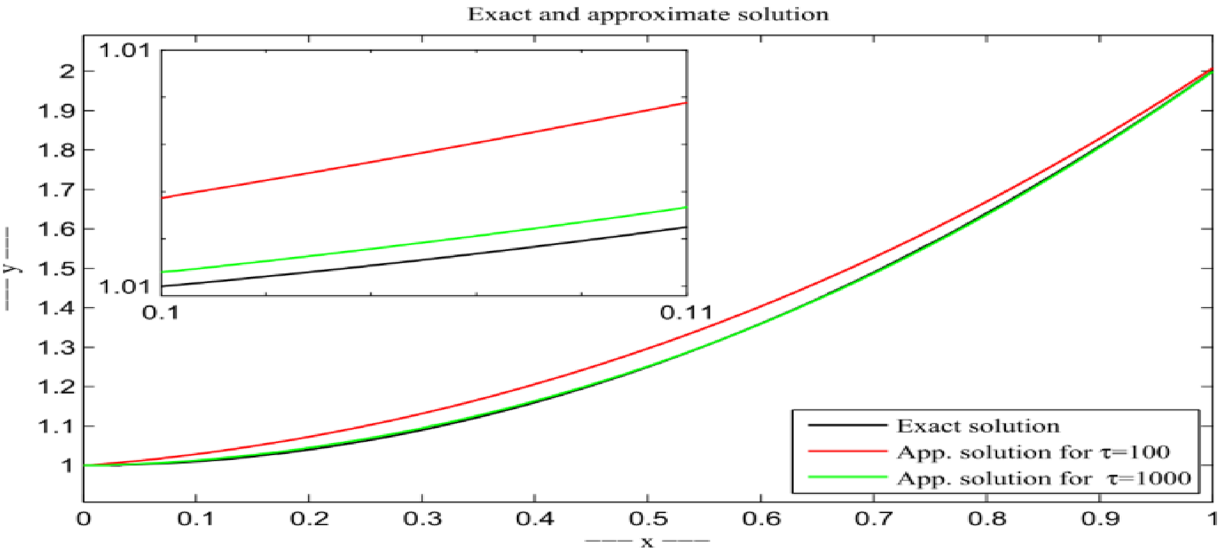


Figure 3: Precise and rough answers to Example 4.1

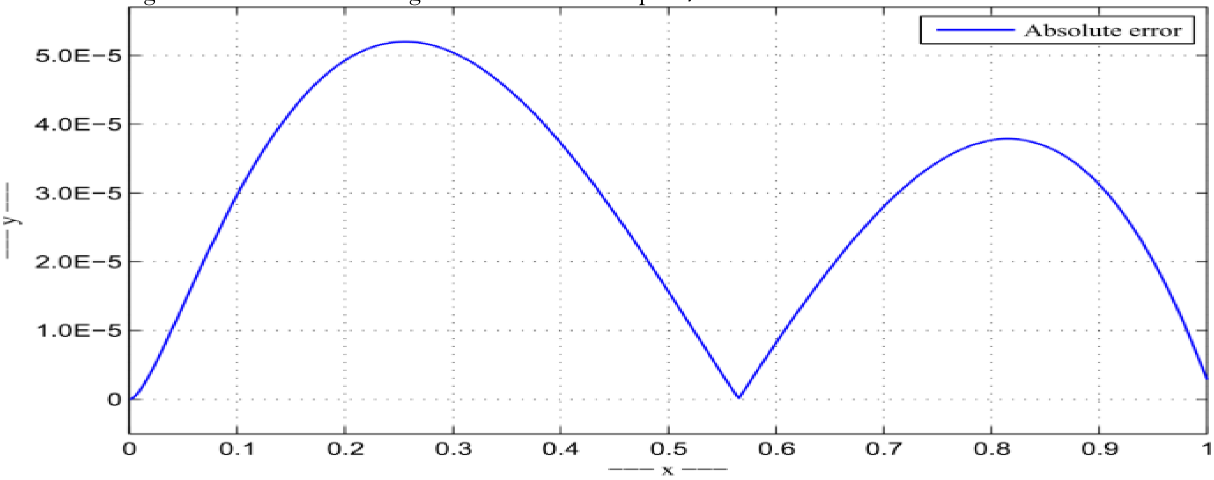


Figure 4: Example 4.1 Absolute Errors

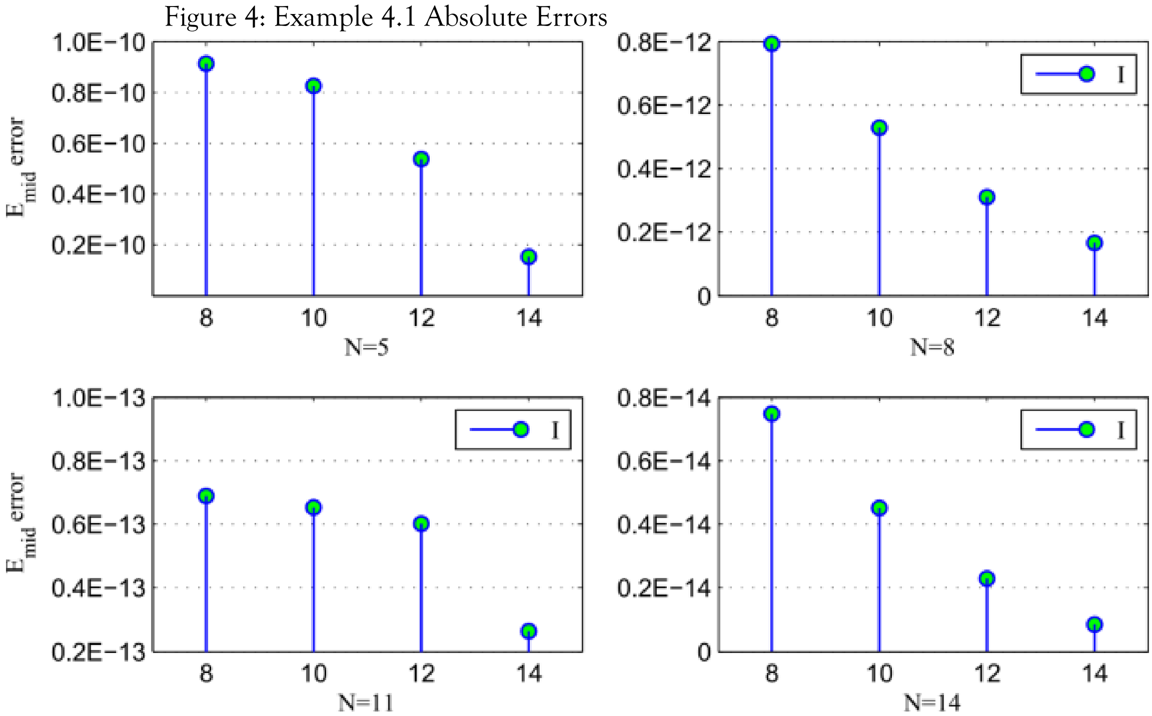


Figure 5 shows how well the ANN design works with Example 4.1.

Keep in mind that the changeable parameters were chosen at random as tiny, positive real integers each time the training process was carried out.

5 CONCLUSION

In this study, we have effectively approximated the higher-order linear fractional Volterra integro-differential problems of the Caputo type using ANN and the PS technique. Some advantageous aspects of the PS technique were used in conjunction with the LMS rule to convert the aforementioned particular a fractional issue into one that involves minimization. An extensive architecture of ANNs were undoubtedly rooted in the recent simulation and modelling of a number of complex real-world occurrences. The error BP technique was used while taking into account minor modifications to the learning process due to the incredibly intricate structure of the observed situation. The challenge of optimisation on designated subdomains was then approximated using the multilayer neural network that was constructed. Two fractional problems were investigated an order to assess an accuracy of the existing numerical method. Comparing numerical results with equally accurate ones for different solution domain partitionings showed the efficiency and reliability of the proposed approach. The most significant outcome of this study is the provision of different orders of fractional derivatives of the kind buried neurons use this activation function. An effective formulation was required to compute fractional derivatives of the sigmoidal function. It is anticipated that this paper will highlight the importance of the suggested approach for solving iterative functions as well as for additional research in related or particular topics. The inadequacies of earlier research may be addressed and novel answers to novel issues may be discovered by extending the suggested approach to a broad class of non-linear scenarios.

6. Future Scope

The future scope of using Artificial Neural Networks (ANNs) to solve fractional higher-order linear integro-differential equations (IHODEs) is vast and promising. There are many exciting directions to explore, ranging from nonlinear models, multi-scale problems, and high-dimensional systems, to integrating the methodology with real-time applications and parallel computing solutions. By expanding the capabilities of ANNs in this domain, significant advancements can be made in both theoretical modeling and practical applications across a wide range of scientific, engineering, and industrial fields.

This methodology, leveraging the strengths of ANNs in handling nonlinearities and high-dimensional systems, has demonstrated promising results in solving fractional equations that are otherwise challenging to approach using traditional methods.

Through the use of ANNs, we are able to approximate the solutions of fractional IHODEs with significant accuracy, bypassing the need for explicit analytical solutions or cumbersome numerical methods. This method offers several advantages, including flexibility in dealing with a wide range of boundary conditions, scalability to higher dimensions, and the ability to model systems with non-smooth or irregular behavior.

The methodology also highlights the potential for interdisciplinary applications, extending to fields such as engineering, biology, finance, and environmental science, where fractional differential equations are often used to model complex, non-local phenomena. The ANN-based approach holds promise for real-time simulations, multi-scale problems, and stochastic systems, making it a versatile tool for addressing modern challenges.

Despite its strengths, there are areas where the methodology could be further improved. These include enhancing training algorithms for faster convergence, developing error estimation techniques, and extending the approach to handle more complex nonlinear, time-dependent, and stochastic fractional IHODEs. Moreover, integrating ANNs with traditional numerical methods and applying them in high-performance computing environments will open up new opportunities for solving large-scale, high-dimensional problems efficiently.

In conclusion, the use of ANNs to solve fractional higher-order linear integro-differential equations is a highly promising area of research. By addressing the challenges and expanding the capabilities of this approach, we can unlock its full potential and make significant contributions to the solution of real-world problems across diverse scientific and engineering disciplines.

In this study, the use of Artificial Neural Networks (ANNs) for solving fractional higher-order linear integro-differential equations (IHODEs) has been demonstrated to be a powerful and efficient method for addressing complex mathematical models that arise in various scientific and engineering fields. The ANN-based approach offers several advantages, including the ability to approximate solutions to fractional IHODEs without the need for traditional analytical solutions or time-consuming numerical methods.

The flexibility of ANNs makes them particularly suitable for solving problems with complex boundary conditions, nonlinearities, and high-dimensional systems. Additionally, the methodology can easily be adapted to handle systems with fractional derivatives and integrals, providing a versatile tool for tackling a broad range of problems in areas such as engineering, physics, finance, biology, and environmental sciences.

Although the current results are promising, there are areas where further research can enhance the methodology. Key areas of future development include improving training algorithms to increase computational efficiency, extending the approach to solve more complex nonlinear and time-dependent fractional IHODEs, and integrating the ANN method with traditional numerical methods for large-scale problems. Moreover, further work on error estimation and the explainability of ANN models will enhance the reliability and transparency of the solutions.

Overall, the application of ANNs to fractional IHODEs represents a promising direction for solving complex, real-world problems. By expanding the capabilities of this approach and addressing its current limitations, ANNs can play an increasingly important role in advancing the solution of fractional differential equations across various domains.

This research demonstrates the potential of Artificial Neural Networks (ANNs) as a robust tool for solving fractional higher-order linear integro-differential equations (IHODEs). The use of ANNs offers significant advantages in terms of flexibility, scalability, and the ability to handle complex systems that are challenging for traditional analytical and numerical methods. By leveraging the power of ANNs, it is possible to approximate solutions to fractional IHODEs efficiently, even in the presence of non-linearity, irregular boundary conditions, and high-dimensional problems.

The approach is highly adaptable and has shown promise in a range of applications, including engineering, physics, biology, finance, and environmental sciences, where fractional differential equations are frequently employed to model complex, real-world phenomena. Furthermore, the ability of ANNs to manage both fractional derivatives and integral terms positions them as an effective tool for addressing a wide array of problems in these fields.

Despite the promising results, there are areas for further improvement. Future research can focus on optimizing the ANN architectures and training algorithms to enhance performance, particularly for large-scale or highly complex problems. Additionally, extending the methodology to more intricate non-linear, time-dependent, and stochastic systems will broaden its applicability and effectiveness. Combining ANNs with traditional numerical methods could also provide solutions that are both computationally efficient and highly accurate.

In conclusion, the use of ANNs to solve fractional IHODEs is a promising and innovative approach that holds the potential to revolutionize the way complex mathematical models are solved in diverse scientific and engineering disciplines. As the methodology continues to evolve, it will undoubtedly contribute to the development of more accurate, efficient, and scalable solutions for fractional problems in the future.

REFERENCES

1. Alkan, S., Hatipoglu, V.: Approximate solutions of Volterra–Fredholm integro-differential equations of fractional order. Tbil. Math. J. 10(2), 1–13 (2017)
2. Bazgir, H., Ghazanfari, B.: Spectral solution of fractional fourth order partial integro-differential equations. Comput. Methods Differ. Equ. 7(2), 289–301 (2019)
3. Bentrcia, T., Mennouni, A.: On the asymptotic stability of a Bresse system with two fractional damping terms: theoretical and numerical analysis. Am. Inst. Math. Sci. 28(1), 580–622 (2023)
4. Caputo, M.: Linear model of dissipation whose Q is almost frequency dependent II. Geophys. J. R. Astron. Soc. 13, 529–539 (1967)
5. Elbeleze, A.A., Kilicman, A., Taib, B.M.: Approximate solution of integro-differential equation of fractional (arbitrary) order. J. King Saud Univ., Sci. 28(1), 61–68 (2016)
6. Graupe, D.: Principles of Artificial Neural Networks, 2nd edn. World Scientific, Singapore (2007)
7. Hamoud, A.A., Ghadle, K.P., Atshan, S.H.: The approximate solutions of fractional integro-differential equations by using modified Adomian decomposition method. Khayyam J. Math. 5(1), 21–39 (2019)
8. Hassoun, M.H.: Fundamentals of Artificial Neural Networks. MIT Press, Cambridge (1995)
9. Jafarian, A., Measoomy Nia, S., Abbasbandy, S.: Artificial neural networks based modeling for solving Volterra integral equations system. Appl. Soft Comput. 27, 391–398 (2015)
10. Jafarian, A., Measoomy Nia, S., Jafari, R.: Solving fuzzy equations using neural nets with a new learning algorithm. J. Adv. Comput. Res. 3(4), 33–45 (2012)
11. Jafarian, A., Mokhtarpour, M., Baleanu, D.: Artificial neural network approach for a class of fractional ordinary differential equation. Neural Comput. Appl. 28(4), 765–773 (2017)

12. Jafariana, A., Measoomy Nia, S.: An application of ANNs on power series method for solving fractional Fredholm type integro-differential equations. Neural Parallel Sci. Comput. 24 (2016)
13. Jafariana, A., Measoomy Nia, S., Golmankhaneh, A.K., Baleanu, D.: Onartificial neural networks approach with new cost functions. Appl. Math. Comput. 339(15), 546–555 (2018)
14. Kurkcu, O.K.: An evolutionary numerical method for solving nonlinear fractional Fredholm–Volterra–Hammerstein integro-differential-delay equations with a functional bound. Int. J. Comput. Math. 99(11), 2159–2174 (2022)
15. Mennouni, A.: Improvement by projection for integro-differential equations. Math. Methods Appl. Sci. (2020). https://doi.org/10.1002/mma.6318
16. Minai, A.A., Williams, R.D.: On the derivatives of the sigmoid. Neural Netw. 6, 845–853 (1993)
17. Nemati, S., Lima, P.M.: Numerical solution of nonlinear fractional integro-differential equations with weakly singular kernels via a modification of hat functions. Appl. Math. Comput. 327(15), 79–92 (2018)
18. Ordokhani, Y., Dehestani, H.: Numerical solution of linear Fredholm–Volterra integro-differential equations of fractional order. World J. Model. Simul. 12(3), 204–216 (2016)
19. Rahimkhania, P., Ordokhani, Y., Babolian, E.: A numerical scheme for solving nonlinear fractional Volterra integro-differential equations. Iran. J. Math. Sci. Inform. 13(2), 111–132 (2018)
20. Ross, B.: Fractional Calculus and Its Applications. Proceedings of the International Conference Held at the University of New Haven. Springer, Berlin (1974)
21. Rostami, F., Jafarian, A.: A new artificial neural network structure for solving high-order linear fractional differential equations. Int. J. Comput. Math. 95(3), 528–539 (2018)
22. Soltani, Z., Jafarian, A.: A new artificial neural networks approach for diagnosing diabetes disease type II. Int. J. Adv. Comput. Sci. Appl. 7(6), 89–94 (2016)
23. Wang, J.: Numerical algorithm for two-dimensional nonlinear Volterra–Fredholm integral equations and fractional integro-differential equations (of Hammerstein and mixed types). Eng. Comput. 38(9), 3548 3563 (2021)
24. Wang, Y., Zhu, L.: Solving nonlinear Volterra integro-differential equations of fractional order by using Euler wavelet method. Adv. Differ. Equ. (2017). https://doi.org/10.1186/s13662-017-1085-6
25. Wei, J., Tian, T.: Numerical solution of nonlinear Volterra integro-differential equations of fractional order by the reproducing kernel method. Appl. Math. Model. 39, 4871–4876 (2015)
26. Yang, A.M., Han, Y., Mang, Y.Z.: On local fractional Volterra integro-differential equations in fractal steady heat, transfer. Therm. Sci. 20, 789–793 (2016)
27. Zhou, D., Zhang, K., Ravey, A., Gao, F., Miraoui, A.: Parameter sensitivity analysis for fractional-order modeling of lithium-ion batteries. Energies 9(3), 1–26 (2016)
28. Ramdas Vankdothu,.Mohd Abdul Hameed "Adaptive features selection and EDNN based brain image recognition on the internet of medical things", Computers and Electrical Engineering , 103 (2022) 108338.
29. Ramdas Vankdothu,.Mohd Abdul Hameed,Ayesha Ameen,Raheem,Unnisa " Brain image identification and classification on Internet of Medical Things in healthcare system using support value based deep neural network" Computers and Electrical Engineering,102(2022) 108196.
30. Ramdas Vankdothu,.Mohd Abdul Hameed" Brain tumor segmentation of MR images using SVM and fuzzy classifier in machine learning" Measurement: Sensors Journal,Volume 24, 2022, 100440 .