Progress in Fractional Differentiation and Applications
An International Journal

http://dx.doi.org/10.18576/pfda/120307

Physics-Informed Neural Networks for Fractional Integro-Differential-Algebraic Equations: A Mesh-Free Framework with Convergence Analysis

Amin M. Elfeky[1], *Adel S. Hussain*[2,3,4,*], *Essa Melfi Aldosari*[5], *Mohammad A. Tashtoush*[6,7], *and Mohamed Sayed Abdellatif*[8]

[1] Department of Mathematics, College of Science and Humanities, Prince Sattam Bin Abdulaziz University, Alkharj, Saudi Arabia
[2] IT Department, Amedi Technical Institutes, University of Duhok Polytechnic, Duhok, Iraq
[3] Department of Administrative and Financial Affairs, Faculty Division, Al-Iraqi University, Baghdad, Iraq
[4] Department of Computer Engineering, Al-Kitab University, Altun, Kupri, Kirkuk, Iraq
[5] Department of Curriculum and Instruction, College of Education in Al-Kharj, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[6] Department of Basic Sciences, AL-Huson University College, AL-Balqa Applied University, Salt, Jordan
[7] Faculty of Education and Arts, Sohar University, Sohar, Oman
[8] Department of Psychology, College of Education, Prince Sattam Bin Abdulaziz University, Alkharj, Saudi Arabia

Received: 20 May 2026, Revised: 1 Jun. 2026, Accepted: 10 Jun. 2026
Published online: 1 Jul. 2026

Abstract: Fractional integro-differential-algebraic equations (FIDAEs) provide a mathematical framework for describing dynamical systems characterized by nonlocal memory effects, hereditary behavior, and algebraic constraints. However, their numerical approximation remains challenging because of the coupling between fractional differential operators, integral terms, and algebraic conditions. This study presents a modified physics-informed neural network (PINN) framework for approximating solutions to FIDAEs, including fractional Bagley–Torvik-type models. The proposed approach incorporates fractional differential operators, Volterra-type integral terms, initial conditions, and algebraic constraints into a unified physics-informed loss function. Numerical quadrature is employed to approximate the nonlocal operators, while neural network optimization is used to minimize the governing equation residuals. The mathematical formulation is accompanied by a convergence analysis that discusses the approximation properties of the proposed framework under regularity and continuity assumptions. The computational performance is investigated through representative fractional integro-differential problems with available reference solutions. Numerical comparisons based on pointwise absolute errors demonstrate that the proposed method can produce accurate approximations and competitive results relative to previously reported numerical approaches. The findings illustrate the potential of physics-informed neural networks as a flexible computational tool for fractional dynamical systems with nonlocal operators and algebraic constraints, while identifying the need for further theoretical and computational investigation of convergence and stability.

Keywords: Fractional Calculus, Fractional Integro-Differential-Algebraic Equations, Physics-Informed Neural Networks, Mesh-Free Method, Bagley–Torvik Equation, Convergence Analysis.

1 Introduction

Recent investigations have demonstrated the applicability of fractional calculus to numerical analysis and the mathematical modeling of complex physical and biological systems. However, these studies differ substantially in their mathematical formulations, research objectives, and approaches to fractional-order operators. [1] investigated a hybrid numerical technique for approximating solutions to fractional-order integro-differential equations, providing a methodological contribution relevant to the numerical treatment of coupled differential and integral operators. In contrast, [2] examined a fractional model of hepatitis B virus dynamics involving a nonlocal kernel, illustrating the use

* Corresponding author e-mail: adel.sufyan@dpu.edu.krd

© 2026 NSP
Natural Sciences Publishing Cor.

of fractional operators in biological modeling. [3] considered the dynamics of a generalized time-fractional viscous-capillarity compressible fluid model, demonstrating the application of fractional formulations to fluid dynamical problems.

Further investigations have addressed fractional mathematical models involving alternative operators and different application settings. [4] studied hepatitis B virus infection using the Hilfer fractional derivative, highlighting the importance of the selected fractional operator in formulating biological dynamical models. [5] investigated a fractional-dimensional formulation associated with calcium and buffer distribution, while [6] considered a fractionalized mathematical investigation of tuberculosis. These studies collectively illustrate the breadth of fractional calculus applications, particularly in systems where nonlocal mathematical formulations are of interest. From a methodological perspective, an important distinction should be made between studies primarily concerned with modeling specific physical or biological phenomena and investigations focused on developing numerical methods for fractional integro-differential equations. [1] is particularly relevant to the numerical approximation aspect of the present research, whereas [2,3,4,5,6] primarily establish the broader applicability of fractional mathematical modeling. However, these contributions should not be interpreted as direct demonstrations of a unified physics-informed neural network formulation for fractional integro-differential-algebraic systems. The present investigation focuses on the additional mathematical and computational challenges associated with coupling fractional differential operators, Volterra-type integral terms, and algebraic constraints within a residual-based neural network framework.

The emergence of various definitions of fractional derivatives in the literature in recent years is due to the increasing popularity of a range of definitions that captivate different physical and mathematical phenomena that it is intended to represent. One of these is the Caputo fractional derivative introduced by [7] which is commonly known to have a wide usage in initial value problems of practical use. Nevertheless, an important weakness of this operator is that it relies on one and only a singular kernel, and it may be a problem its analytical tractability as well as numerical implementation. In order to overcome this weakness Caputo and Fabrizio developed a non-singular fractional derivative using an exponential kernel [8]. In as much as this modification removes the singularity, limitations are imposed in that the exponential kernel may not be appropriate to processes that are not characterized by exponential decays. Atangana and Baleanu proposed a more generalized form that involves the kernel in terms of the ousted Mittag-Leffler function that gives rise to a non-singular fractional derivative [9,10]. Based on this model, Refai and Baleanu have recently suggested the Modified Atangana Baleanu- Capito (MABC) derivative [11]. It maintains the beneficial feature of its predecessors but has greater flexibility and the descriptive ability of non-trivial systems, which could not be effectively represented with the original Atangana-Baleanu derivatives. In [12], generalized theoretical studies of MABC derivative have been considered. Numerical illustrations of the practical uses of the MABC derivative are the finite difference method in the advection -dispersion equations [13], discrete Chebyshev poly approximation [14] and in the solution of nonlinear partial differential equations (NPDEs) [15]. The theory has been applied in many fields of physics, electronics, information technology, computer science, fiber optics, mechanics, electrical engineering, and plasma physics, as well as nonlinear optics and communication systems, among many others, all primarily based on nonlinear PDEs [16,17,18,19,20,21,22,23,24].

Fractional differential equations have been extensively employed to describe anomalous transport, viscoelastic behavior, and dynamical processes involving memory and hereditary effects. In particular, fractional advection–dispersion equations (FADEs) extend classical transport models by incorporating nonlocal operators capable of representing deviations from conventional Fickian diffusion. However, the numerical treatment of such models remains challenging because fractional derivatives depend on the solution history and may involve singular or weakly singular kernels. These difficulties become more pronounced when fractional differential operators are coupled with integral terms and algebraic constraints, motivating the development of flexible and reliable computational methods for fractional integro-differential-algebraic systems. But empirical and theoretical evidence is mounting to suggest that the actual contaminant transport in porous media typically cannot be described by this assumption; instead, it has non-Fickian or non-anomalous behavior which cannot be represented satisfactorily with traditional ADEs. In order to overcome these drawbacks, it has been suggested that fractional advection-dispersion equations (FADEs) should be used as a more effective alternative. FADEs are also characterized by the addition of the anomalous diffusion and memory effects through the use of fractional derivatives, which puts them in a better position to represent the transport processes in more realistic complex systems in the subsurface. The approximation of such models is also subject to great mathematical difficulties, as opposed to the difficulties of their numerical solution, especially under realistic initial and boundary conditions. To achieve the accurate and efficient solution of FADEs, special numerical tools are hence needed. [25] have investigated the uniqueness and existence of solutions to some initial and boundary value equations of fractional advection-dispersion equations.

© 2026 NSP
Natural Sciences Publishing Cor.

Physics-informed neural networks (PINNs) have emerged as a flexible computational framework for approximating solutions to differential equations by incorporating governing equations, initial and boundary conditions, and physical constraints into a residual-based loss function. Their application has been extended beyond classical differential equations to fractional-order models involving nonlocal operators and memory-dependent behavior. In particular, [24] introduced fractional physics-informed neural networks (fPINNs) for space-time fractional advection–diffusion equations, establishing an important foundation for the integration of fractional differential operators into neural network formulations. Subsequent investigations have developed PINN-based approaches for Caputo fractional differential equations [25], high-dimensional fractional and tempered fractional partial differential equations [26], and spectral approximations of fractional operators [27]. These developments demonstrate that fractional PINNs constitute an established and actively developing research area. Nevertheless, the simultaneous treatment of fractional differential operators, Volterra-type integral terms, and algebraic constraints introduces additional mathematical and computational challenges that motivate further investigation.

Recent advance in scientific machine learning have significantly expanded the applicability of physics-informed neural networks (PINNs) to fractional differential equations and computational models governed by nonlocal operators. The pioneering fractional physics-informed neural network (fPINN) framework introduced by [24] combined neural network approximation with numerical discretization of fractional differential operators to address space-time fractional advection-diffusion equations. Subsequently, [25] developed an optimization-based PINN methodology for differential equations involving Caputo fractional derivatives, employing constrained expressions to incorporate prescribed conditions into the approximation. [26] investigated the computational challenges associated with fractional and tempered fractional partial differential equations, with particular emphasis on high-dimensional problems and Monte Carlo-based approximation of fractional operators. More recently, [27] introduced a spectrally adapted PINN approach for fractional partial differential equations, demonstrating an alternative strategy for addressing nonlocal operators on unbounded domains. Collectively, these investigations highlight the increasing importance of neural-network-based numerical methods for fractional dynamical systems while also demonstrating the continuing challenges associated with nonlocal operator evaluation, computational efficiency, and numerical accuracy.

In addition to advances in the numerical approximation of fractional operators, recent research has emphasized the mathematical reliability and computational performance of physics-informed neural networks. [28] examined theoretical and numerical aspects of PINN-based approximation, highlighting the distinction between approximation, generalization, and optimization errors. Related investigations have considered PINN formulations for differential-algebraic systems [29], while comprehensive reviews have discussed the influence of network architecture, collocation strategies, loss-function design, and optimization procedures on numerical performance [30]. These studies provide an important foundation for investigating PINN formulations for fractional integro-differential-algebraic equations, particularly with respect to residual consistency, constraint enforcement, and convergence analysis. Recent numerical analyses [28] have emphasized the distinction between approximation, generalization, and optimization errors, together with the essential role of regularity assumptions and stability estimates in establishing convergence properties. Moreover, recent investigations of physics-informed neural networks for differential-algebraic equations have demonstrated the potential of combining neural approximations with structure-preserving numerical techniques to handle coupled dynamical equations and algebraic constraints [31]. Comprehensive reviews of PINN methodologies [32] have further identified the influence of network architecture, loss-function design, collocation strategies, and training procedures on numerical performance. These developments provide important theoretical and computational motivation for extending physics-informed learning frameworks to fractional integro-differential-algebraic systems, in which memory-dependent fractional operators, integral contributions, and algebraic constraints must be treated simultaneously.

The so-called fractional integro-differential-algebraic equations (FIDAE) represent a flexible mathematical model of the complex dynamical systems with memory effects, hereditary properties, and algebraic constraints. They are commonplace in a broad field of research, such as viscoelastic materials, electrical circuits involving fractional elements, fluid flow in porous media and reliability engineering. This is because the presence of fractional derivatives permits FIDAE to describe the nonlocal temporal dynamics of the process, and the terms of integral terms integrate cumulative or history effects [33,34,35]. At the same time, the algebraic components put restrictions on the system state, which characterizes physical or geometrical restrictions. Collectively, these properties produce a very versatile, although mathematically difficult, system of describing real world phenomena. FIDAE is an equation that analytically or numerically is not always solved (particularly when in a nonlinear, high-dimensional or in an initial and boundary condition that is realistic) [36]. Conventional numerical techniques, including finite difference schemes, spectral collocation techniques or operational matrix techniques can be computationally expensive and can need a careful discretization of both fractional and integral operators. Physics-Informed Neural Networks (PINNs) have recently become a strong computational model of solving differential and integro-differential equations, even those of fractional

© 2026 NSP
Natural Sciences Publishing Cor.

order [37]. PINNs make use of deep neural networks to estimate the unknown solution, at the same time imposing the governing equations, initial conditions, and any constraint on the solution directly into the loss. In the case of FIDAE, the derivatives and integrals can be replaced by suitable numerical approximations (e.g. Caputo derivative or Riemann-Liouville derivative), and the integrals can be solved by quadrature or spectral methods [38,39]. The network is optimized by reducing the residual of the FIDAE at collocation points and thus, reduces the solution process to an optimization problem. This method has not only the advantage of avoiding fine meshing but also smooth, high-fidelity approximations that can effectively solve nonlinear, nonlocal and constrained systems. The PINNs have shown strong ability in solving various fractional and integro-differential problems, providing feasible computational approach to the numerical solution to FIDAE in the complex world [40,41].

Despite the substantial progress in fractional physics-informed neural networks, the numerical approximation of coupled fractional integro-differential-algebraic equations (FIDAEs) remains mathematically challenging. In these systems, fractional derivatives introduce nonlocal memory effects, Volterra-type integral operators account for cumulative contributions, and algebraic constraints restrict the admissible solution space. Their simultaneous treatment requires careful approximation of nonlocal operators, consistent enforcement of algebraic conditions, and appropriate balancing of the residual components during neural network training. Furthermore, small residual losses alone do not guarantee convergence to the exact solution without suitable consistency and stability assumptions. Motivated by these challenges, the present study investigates a modified PINN framework for the numerical approximation of FIDAEs, with emphasis on the formulation of the differential–integral and algebraic residuals, numerical accuracy, and the mathematical conditions relevant to convergence.

This paper suggests a modified PINN framework to solve fractional integro-differential-algebraic equation. The key features of our method are better accuracy and efficiency through architecture design, training components, and training regime to capture the behavior of the fractional derivative more effectively and the typical pitfalls of slow convergence, boundary layer errors, or instability [24]. Various analytical, numerical, and computational approaches have been employed in the literature to investigate a wide range of related problems; see [41,42,43].

The principal contribution of this study is the formulation and numerical investigation of a modified physics-informed neural network (PINN) framework for fractional integro-differential-algebraic equations. The proposed approach incorporates fractional differential operators, Volterra-type integral terms, initial conditions, and algebraic constraints into a unified residual-based loss formulation. Numerical approximation techniques are employed to evaluate the nonlocal fractional and integral operators, while the neural network parameters are optimized to minimize the governing equation residuals and constraint violations. The methodology is investigated through representative numerical examples, and its approximation accuracy is assessed using analytical reference solutions and comparisons with previously reported results. This formulation provides a computational approach for investigating fractional dynamical systems involving memory effects and coupled algebraic constraints.

This analysis aims to process the following FIDAE in general fractional integro-differential-algebraic equation (FIDAE) can be written as:

This analysis aims to process the following FIDAE in general fractional integro-differential-algebraic equation (FIDAE) can be written as:

$$\begin{cases} {}^{C}D_t^{\alpha}x(t) = f\left(t, x(t), y(t), \int_0^t K(t,s)x(s)ds\right) \\ 0 = g\left(t, x(t), y(t), \int_0^t L(t,s)x(s)ds\right) \\ x(0) = x_0 \end{cases} \tag{1}$$

Where ${}^{C}D_t^{\alpha}(\cdot)$ denotes the Caputo fractional derivative of order α, $f(\cdot)$ represents the fractional differential operator, and $g(\cdot)$ denotes the algebraic constraint. The functions $K(t,s)$ and $L(t,s)$ are memory kernels, which may take various forms such as power-law, exponential, or Mittag-Leffler kernels, accounting for nonlocal and hereditary effects. The vector x_0 specifies the prescribed initial condition. In this case, $x(t)$ is the state variable that characterizes the system development. The time-dependent derivative ${}^{C}D_t^{\alpha}x(t)$ of the object is the Caputo fractional derivative of α a non-zero order, which considers nonlocal temporal effect and hereditary effects. The integral term

$$\int_0^t K(t,s)x(s)ds. \tag{2}$$

is a Volterra-type operator with kernel $K(t,s)$, which is a cumulative effects and long-term memory of states in the past. It is the algebraic constraint of admissible states provided by the algebraic expression $F(\cdot) = 0$ that connects constituents of these algebraic, differential, and integral together as one.

© 2026 NSP
Natural Sciences Publishing Cor.

The ability that we use to approximate the solution of the proposed model is the Physics-Informed Neural Networks (PINNs) that have recently been developed [42,43]. In this method, the neural network structure exclusively involves the underlying physical laws in the loss coding. The Caputo fractional derivatives properties are used to create a customized loss model to explain the nonlocal and memory-dependent properties of the system. The governing equations are applied at a few collocation points along with the initial and boundary conditions and the problem is reformulated as an optimization task. The nonlinear equations resulting are then solved by the use of numerical optimization methods which generate accurate and computationally efficient solutions.

1.1 Fractional Operators (Preliminaries)

As explained in the above section it is yet to be agreed on a physical or geometrical interpretation of the fractional derivatives, and this has led to the birth of a number of alternative definitions in the literature. These underlying concepts need to be introduced, as they are particularly important in defining and supporting the theoretical framework that will be used in the current research.

Definition 1 (Caputo Fractional Derivative) [9,10].

Let $\alpha > 0$, and let $n \in \mathbb{N}$ satisfy $n-1 < \alpha < n$. Suppose that $u^{(n-1)}$ is absolutely continuous on $[0,T]$. The Caputo fractional derivative of order α is defined by

$$ {}^{C}D^{\alpha}_{0+}u(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-s)^{n-\alpha-1} u^{(n)}(s)ds. \tag{3}$$

For an integer order $\alpha = n$, the Caputo derivative coincides with the classical derivative:

$$ {}^{C}D^{n}_{0+}u(t) = u^{(n)}(t). \tag{4}$$

Under suitable regularity assumptions, an equivalent representation is

$$ {}^{C}D^{\alpha}_{0+}u(t) = D^{\alpha}_{0+}\left[u(t) - \sum_{k=0}^{n-1} \frac{u^{(k)}(0)}{k!} t^k\right], \tag{5}$$

where D^{α}_{0+} denotes the Riemann-Liouville fractional derivative. The corresponding initial conditions are

$$ u^{(k)}(0) = u_k, \; k = 0,1,\ldots,n-1. \tag{6}$$

For $0 < \alpha < 1$, the definition reduces to

$$ {}^{C}D^{\alpha}_{0+}u(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} u'(s)ds. \tag{7}$$

For $1 < \alpha < 2$, the corresponding expression is

$$ {}^{C}D^{\alpha}_{0+}u(t) = \frac{1}{\Gamma(2-\alpha)} \int_0^t (t-s)^{1-\alpha} u''(s)ds. \tag{8}$$

Definition 2. [12] Let $f(t)$ be a function defined on $[0,T]$ and let $\alpha > 0$. The Riemann-Liouville fractional integral of order α is defined by

$$ \left(I^{\alpha}_{t} f\right)(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s)ds, \; t > 0, \tag{9}$$

where $\Gamma(\cdot)$ denotes the Gamma function.

2 Methodology

Physics-Informed Neural Networks (PINNs) are used here to solve Bagley physical equations of third-type. Fractional integro-differential-algebraic equations are of type fractional differential algebraic equations, which are directly integrated with the fractional derivatives.

© 2026 NSP
Natural Sciences Publishing Cor.

2.1 Modified Physics-Informed Neural Networks (PINNs)

The proposed computational framework employs a modified physics-informed neural network (PINN) formulation to approximate the solutions of fractional integro-differential-algebraic equations, including fractional Bagley-Torvik-type models. The governing fractional differential equations, associated integral operators, and algebraic constraints are incorporated into the physics-informed loss formulation. The nonlocal fractional and integral terms are evaluated using suitable numerical approximation procedures, while the trainable parameters of the neural network are determined by minimizing the resulting residual-based objective function. The framework is intended to provide a flexible numerical approximation of fractional dynamical systems with memory-dependent behavior and algebraic restrictions. Its numerical performance is assessed through comparisons with available reference solutions using appropriate error measures [25].

The numerical implementation of the proposed physics-informed neural network (PINN) framework employs a fully connected feed-forward neural network to approximate the unknown solutions of the considered fractional integro-differential-algebraic equations. The baseline architecture consists of two hidden layers containing 6 and 9 neurons, respectively. The independent variable is introduced through the input layer, while the output layer represents the approximate solution of the governing system. The selection of a relatively compact architecture is motivated by the low-dimensional nature of the numerical test problems and the objective of reducing computational complexity. Nevertheless, the adequacy of such an architecture cannot be established solely on the basis of its size. Its performance must be evaluated with respect to the regularity and complexity of the target solutions, the accuracy of the fractional operator approximation, and the numerical enforcement of the governing equations and algebraic constraints.

To ensure reproducibility, the implementation is characterized by the activation function, parameter initialization method, optimizer, initial learning rate, learning rate scheduling strategy, number and distribution of collocation points, maximum number of training epochs, and stopping criterion. The same information is summarized in Table 3. To assess the suitability of the baseline architecture, a systematic sensitivity analysis is conducted by varying the number of hidden layers and neurons while retaining consistent numerical approximation and optimization settings. The resulting configurations are evaluated against the available analytical solutions to determine the influence of network capacity on prediction accuracy, convergence behavior, computational time, and numerical stability.

The analysis is intended to establish whether the baseline configuration provides a reasonable balance between approximation accuracy and computational efficiency for the numerical examples considered in this study, rather than claiming that this architecture is universally optimal for nonlinear fractional integro-differential-algebraic systems. The proposed physics-informed neural network employs a fully connected feed-forward architecture consisting of two hidden layers with 6 and 9 neurons, respectively. This relatively compact architecture was initially selected to investigate the numerical approximation of the low-dimensional fractional integro-differential-algebraic benchmark problems considered in the present study. The use of a compact network can reduce the number of trainable parameters and the computational effort required during optimization.

Nevertheless, the approximation capability of a neural network depends not only on the number of hidden layers and neurons but also on the regularity and complexity of the target solution, the activation functions, the accuracy of the fractional operator approximation, and the numerical optimization procedure. Consequently, the relatively small architecture adopted in this study should be regarded as a baseline computational configuration rather than a generally optimal architecture for fractional integro-differential-algebraic equations.

For more complicated systems exhibiting strong nonlinearities, rapid temporal variations, stiffness, or highly oscillatory behavior, the expressive capacity of a compact network may become insufficient. In such situations, increasing network depth or width, selecting suitable activation functions, and employing adaptive collocation strategies may be necessary to improve approximation accuracy.

The architecture sensitivity analysis described in Section 3.5 provides a framework for assessing these effects through comparisons between networks of different depths and widths. However, quantitative conclusions regarding the relative performance of these architectures require completed numerical experiments under consistent training conditions. Accordingly, the selected two-hidden-layer architecture is not claimed to be universally optimal, and its applicability beyond the benchmark problems considered remains subject to further numerical validation. The predictions of the network were tested against the target values using the Mean squared error (MSE) loss function and parameter (weights and biases) of the model were successively optimized using the gradient method to achieve the desired performance level.

Differential operators in fractional integro-differential-algebraic equation have nonlocal and memory-dependent properties, which add an extra complexity to the application of Physics-Informed Neural Networks (PINNs). The loss functional can be adapted to deal with these problems by simply adding a fractional derivative in the loss functional such that the solution found, both in terms of the governing equations and the initial or boundary conditions. PINNs provide an effective and efficient tool to solve partial differential equations (PDEs), fractional integro-differential-algebraic equation, and similar models, in which traditional numerical algorithms might need a lot of resources and computation

© 2026 NSP
Natural Sciences Publishing Cor.

time. For the considered fractional integro-differential-algebraic equation (FIDAE) system, let $u_\theta(t)$ denote the neural network approximation, where θ represents the trainable network parameters. The differential-integral residual is defined as

$$R_f(t;\theta) = {}^{C}D^{\alpha}_{0+}u_\theta(t) - F\left(t, u_\theta(t), \int_0^t K(t,s)u_\theta(s)ds\right). \tag{10}$$

The algebraic residual is defined by

$$R_g(t;\theta) = G(t, u_\theta(t)). \tag{11}$$

Accordingly, the mean-square differential-integral residual loss is given by

$$L_f = \frac{1}{N_f}\sum_{i=1}^{N_f} \left\|R_f(t_i;\theta)\right\|^2. \tag{12}$$

Similarly, the algebraic constraint loss is

$$L_g = \frac{1}{N_g}\sum_{j=1}^{N_g} \left\|R_g(t_j;\theta)\right\|^2. \tag{13}$$

The initial-condition loss is expressed as

$$L_{IC} = \left\|u_\theta(0) - u_0\right\|^2. \tag{14}$$

The standard PINN formulation employs an unweighted combination of the differential-integral residual, algebraic residual, and initial-condition loss:

$$L_{\text{standard}} = L_f + L_g + L_{IC}. \tag{15}$$

In this configuration, all loss components are assigned unit weights. The modified loss formulation is combined with fixed positive weighting coefficients:

$$L_{\text{fixed}} = \lambda_f \widetilde{L}_f + \lambda_g \widetilde{L}_g + \lambda_{IC}\widetilde{L}_{IC}. \tag{16}$$

Here, $\lambda_f, \lambda_g, \lambda_{IC} > 0$ are constant weighting coefficients throughout training. The terms $\widetilde{L}_f, \widetilde{L}_g$, and $\widetilde{L}_{IC}$ denote the corresponding components of the modified loss formulation. The modified loss formulation is combined with dynamically adjusted weighting coefficients:

$$L^{(k)}_{\text{adaptive}} = \lambda_f^{(k)}\widetilde{L}_f + \lambda_g^{(k)}\widetilde{L}_g + \lambda_{IC}^{(k)}\widetilde{L}_{IC}. \tag{17}$$

Here, k denotes the optimization iteration, while $\lambda_f^{(k)}, \lambda_g^{(k)}$, and $\lambda_{IC}^{(k)}$ represent the adaptive weighting coefficients updated during training. A typical PINN framework is described in the following step-by-step manner [26-28]:

Input:

–Differential equation operator $L(\cdot)$, source term $f(t)$
–Domain $[a,b]$
–Boundary/Initial conditions $u(t_i)$ at $t_i \in$ (boundary, initial points)
–Network architecture parameters (layers, neurons, activation functions)
–Learning parameters (optimizer, learning rate α, tolerance ε, max epochs)

Output:

–Trained neural network $u_{NN}(t;\theta)$ approximating solution $u(t)$.

© 2026 NSP
Natural Sciences Publishing Cor.

2.2 Formal Algorithm for the Proposed FIDAE-PINN Framework

The computational algorithm is specifically formulated for fractional integro-differential-algebraic equations (FIDAEs), where fractional differential operators, nonlocal Volterra integral terms, and algebraic constraints must be treated simultaneously. Unlike a conventional physics-informed neural network (PINN) designed solely for differential equations, the present formulation distinguishes between differential state variables and algebraic variables and incorporates their associated residuals into the training procedure. Consider a representative FIDAE system of the form

$$ {}^{C}D_{0+}^{\alpha}x(t) = F(t,x(t),z(t),V[x](t)), \tag{18}$$

subject to the algebraic constraint

$$ G(t,x(t),z(t)) = 0, \tag{19}$$

where the Volterra integral operator is defined by

$$ V[x](t) = \int_0^t K(t,s)x(s)ds. \tag{20}$$

Here, $x(t)$ denotes the differential state, $z(t)$ denotes the algebraic variable, and $\alpha > 0$ is the fractional order. Initial conditions are prescribed according to the order of the Caputo derivative and must be compatible with the algebraic constraints.

Step 1: Specify the FIDAE Model and Consistency Conditions. Identify the differential equations, algebraic constraints, Caputo orders, Volterra kernels, computational interval, and required initial conditions. Verify the consistency of prescribed initial data with the algebraic equations. In particular, when $G(t,x,z) = 0$ is imposed at $t = 0$, ensure that the initial states satisfy

$$ G(0,x_0,z_0) = 0. \tag{21}$$

Step 2: Construct the Neural Network Approximation. Define neural network outputs for the differential and algebraic variables:

$$ (x_\theta(t),z_\theta(t)) = N_\theta(t). \tag{22}$$

The baseline architecture considered in the manuscript consists of two hidden layers containing six and nine neurons, respectively. The network parameters θ include the trainable weights and biases. If the algebraic variables can be eliminated analytically, an equivalent reduced formulation may be considered, provided the reduction is mathematically valid.

Step 3: Discretize the Fractional and Volterra Operators. Select collocation nodes and evaluate the Caputo derivative and Volterra integral operator using the fractional discretization and quadrature procedures described in Section 3.1, Step 3. The discretization must be consistent with the fractional order and the regularity of the kernel. Denote the resulting discrete operators by

$$ D_h^{\alpha}x_\theta[t_i] \quad \text{and} \quad V_h[x_\theta](t_i). \tag{23}$$

Step 4: Construct the Differential-Integral Residual. At each differential collocation point t_i, evaluate

$$ r_{f,\theta}(t_i) = D_h^{\alpha}x_\theta(t_i) - F\left(t_i,x_\theta(t_i),z_\theta(t_i),V_h[x_\theta](t_i)\right). \tag{24}$$

This residual enforces the governing fractional differential-integral equation.

Step 5: Enforce the Algebraic Constraints. Evaluate the algebraic residual independently at the algebraic collocation points:

$$ r_{g,\theta}(t_i) = G(t_i,x_\theta(t_i),z_\theta(t_i)). \tag{25}$$

The corresponding algebraic constraint loss is

$$ L_g(\theta) = \frac{1}{N_g}\sum_{i=1}^{N}\left\|r_{g,\theta}(t_i)\right\|^2. \tag{26}$$

© 2026 NSP
Natural Sciences Publishing Cor.

This term penalizes violations of the algebraic equations throughout the training domain. The algebraic residual is not replaced by a generic boundary-condition loss because it represents a pointwise condition coupling the differential and algebraic states. For systems in which the algebraic equation can be solved explicitly, an alternative approach is to parameterize the algebraic variable as

$$z_\theta(t) = \phi(t, x_\theta(t)). \tag{27}$$

Such an approach enforces the algebraic relation by construction but requires a well-defined and numerically stable algebraic solution mapping.

Step 6: Incorporate Initial Conditions and Construct the Total Loss. Define the fractional differential-integral loss by

$$L_f(\theta) = \frac{1}{N_f} \sum_{i=1}^{N} \left\| r_{f,\theta}(t_i) \right\|^2. \tag{28}$$

The initial-condition penalty is formed from the required initial values and, for higher fractional orders, the corresponding integer-order initial derivatives. For example, when $0 < \alpha < 1$,

$$L_{IC}(\theta) = \left\| x_\theta(0) - x_0 \right\|^2. \tag{29}$$

For $n-1 < \alpha < n$, the initial-condition loss can be generalized to

$$L_{IC}(\theta) = \sum_{k=0}^{n-1} \left\| x_\theta^{(k)}(0) - x_k \right\|^2. \tag{30}$$

The overall physics-informed loss is

$$L_{\text{total}}(\theta) = \lambda_f L_f(\theta) + \lambda_g L_g(\theta) + \lambda_{IC} L_{IC}(\theta). \tag{31}$$

Here, $\lambda_f > 0, \lambda_g > 0$ and $\lambda_{IC} > 0$ denote the weighting coefficients associated with the differential-integral residual, algebraic constraint, and initial condition penalties, respectively. Their numerical values and any adaptive update mechanism must be specified according to the actual computational implementation.

Step 7: Optimize the Network Parameters. Minimize the total physics-informed loss with respect to θ using the selected optimization method. At each optimization iteration, evaluate the discrete nonlocal operators, construct the differential-integral and algebraic residuals, calculate the loss components, and update the network parameters. The optimization problem is expressed as

$$\theta^* = \arg\min_\theta L_{\text{total}}(\theta). \tag{32}$$

Monitor the evolution of L_f, L_g, and L_{IC} separately to identify whether training difficulties arise from the fractional differential equation, the algebraic constraints, or the initial conditions.

Step 8: Validate the Trained Solution. Evaluate the trained neural network at independent testing points. When an analytical solution is available, calculate the root mean square error (RMSE), mean absolute error (MAE), and maximum absolute error. Independently evaluate the algebraic constraint violation using

$$E_g = \max_{t_i \in T_{\text{test}}} \left\| G(t_i, x_\theta(t_i), z_\theta(t_i)) \right\|. \tag{33}$$

The final approximation should be assessed in terms of solution accuracy, differential-integral residuals, and algebraic consistency. A small aggregate loss alone does not guarantee that each individual constraint has been adequately satisfied.

3 Applications

In this case, a Physics-Informed Neural Network (PINN) version is used to obtain a numerical solution of a fractional integro-differential-algebraic equation of the form in Eq. (1). The fractional PINN algorithm was applied to the following solution framework.

© 2026 NSP
Natural Sciences Publishing Cor.

3.1 Algorithm Steps

Step 1: Problem Definition

–Specify the fractional integro-differential-algebraic equation (or differential-algebraic system).
–Define the fractional order α, time domain $t \in [0,T]$, and source term $h(t)$.
–Prescribe initial and/or boundary conditions

Step 2: Neural Network Construction

–Construct a feed-forward neural network with input variable t.
–Define the network output as:
  –$y_\theta(t)$ for differential equations, or
  –$\left(t, x(t), {}^{C}D_t^{\alpha}x(t), \int_0^t K(t,s)x(s)ds\right)$ for integro-differential-algebraic systems.
–Initialize all weights and biases randomly.

Step 3: Numerical Approximation of Fractional and Integral Operators The numerical approximation of the fractional differential and Volterra integral operators constitutes an essential component of the proposed physics-informed neural network framework. To obtain a reproducible formulation, the nonlocal operators are discretized on a temporal grid

$$t_n = nh, \quad h = \frac{T}{N}, \quad n = 0,1,\ldots,N, \tag{34}$$

where h denotes the uniform time step and N is the number of temporal subintervals. For a Caputo derivative of order $0 < \alpha < 1$, a possible discretization is the classical L1 approximation, given by

$${}^{C}D_{0+}^{\alpha}u_\theta(t_n) \approx \frac{1}{h^{\alpha}\Gamma(2-\alpha)} \sum_{k=0}^{n-1} a_k^{(\alpha)} \left[u_\theta(t_{n-k}) - u_\theta(t_{n-k-1})\right]. \tag{35}$$

The corresponding fractional convolution weights are defined by

$$a_k^{(\alpha)} = (k+1)^{1-\alpha} - k^{1-\alpha}, \quad k = 0,1,\ldots,n-1. \tag{36}$$

For fractional orders $1 < \alpha < 2$, the Caputo derivative involves the second derivative of the solution and is defined as

$${}^{C}D_{0+}^{\alpha}u_\theta(t) = \frac{1}{\Gamma(2-\alpha)} \int_0^t (t-s)^{1-\alpha} u_\theta''(s)ds. \tag{37}$$

In this case, a product-integration approximation based on increments of the first derivative can be written as

$${}^{C}D_{0+}^{\alpha}u_\theta(t_n) \approx \frac{h^{1-\alpha}}{\Gamma(3-\alpha)} \sum_{k=0}^{n-1} b_k^{(\alpha)} \left[u_\theta'(t_{n-k}) - u_\theta'(t_{n-k-1})\right]. \tag{38}$$

The corresponding fractional weights are

$$b_k^{(\alpha)} = (k+1)^{2-\alpha} - k^{2-\alpha}, \quad k = 0,1,\ldots,n-1. \tag{39}$$

The neural network derivatives appearing in this approximation can be evaluated through automatic differentiation. The weakly singular kernels associated with the Caputo derivative are treated by analytically integrating the kernel contributions over individual temporal subintervals. This avoids direct evaluation of the singular kernel at its endpoint and produces explicit fractional convolution weights. The Volterra integral operator is expressed as

$$\mathscr{V}[u_\theta](t_n) = \int_0^{t_n} K(t_n,s)u_\theta(s)ds. \tag{40}$$

For a sufficiently smooth integrand, it may be approximated using the composite trapezoidal rule:

$$\mathscr{V}_h[u_\theta](t_n) = h\left[\frac{1}{2}K(t_n,t_0)u_\theta(t_0) + \sum_{j=1}^{n-1} K(t_n,t_j)u_\theta(t_j) + \frac{1}{2}K(t_n,t_n)u_\theta(t_n)\right]. \tag{41}$$

The corresponding quadrature nodes are

$$s_j = t_j = jh, \quad j = 0,1,\ldots,n, \tag{42}$$

© 2026 NSP
Natural Sciences Publishing Cor.

and the quadrature weights are defined as

$$w_{n,j} = \begin{cases} \frac{h}{2}, & j = 0 \text{ or } j = n, \\ h, & 1 \le j \le n-1. \end{cases} \tag{43}$$

For weakly singular Volterra kernels, a suitable product-integration procedure is required instead of directly applying the standard trapezoidal rule. The resulting approximations of the fractional derivatives and Volterra integral operators are incorporated into the differential-integral residual of the PINN. The network parameters are then optimized by minimizing the governing-equation residuals together with the initial-condition and algebraic-constraint penalties. The fractional orders, temporal resolution, quadrature procedures, and numerical implementation parameters must be specified for each benchmark problem to ensure that the results can be independently reproduced. The fractional differential operator and Volterra integral term appearing in the governing equation are replaced by their corresponding numerical approximations. Accordingly, the differential-integral residual is evaluated using the discretized nonlocal operators and the neural network predictions at the prescribed temporal nodes. The algebraic residual is constructed independently from the governing algebraic constraints. Both residual components are subsequently incorporated into the physics-informed loss function. The differentiation and numerical quadrature procedures are implemented consistently within the computational framework so that gradients of the discrete loss with respect to the network parameters can be evaluated during optimization.

Step 4: Residual Construction

–Compute the fractional integro-differential-algebraic equation residual at collocation points:

$$R_{diff}(\theta) = {}^{C}D_t^{\alpha} x_\theta(t) + y_\theta(t) \tag{44}$$

–If algebraic constraints are present, compute the algebraic residual:

$$R_{alg}(\theta) = \Phi\left(t, x(t), {}^{C}D_t^{\alpha} x(t), \int_0^t K(t,s)x(s)ds\right) \tag{45}$$

Step 5: Loss Function Assembly

–Define the physics-informed loss function as a weighted sum of:
  –Differential equation residual loss
  –Algebraic constraint residual loss (if applicable)
  –Boundary and initial condition loss
–Balance the loss terms using adaptive or fixed weighting coefficients.

Step 6: Prediction After training, the optimized neural network $N(t;\theta^*)$ serves as the approximate solution of $y(t)$.

Step 7: Validation and Error Analysis Evaluate the predictive accuracy using the Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |w(t_i) - N(t_i;\theta^*)|^2} \tag{46}$$

Compare the obtained results with available analytical or numerical reference solutions to assess accuracy and convergence.

3.2 Convergence Analysis

Theorem (Residual-Based Stability of the Fractional PINN Approximation).

Consider the fractional integro-differential-algebraic system

$$\begin{cases} {}^{C}D_{0+}^{\alpha} x(t) = F(t, x(t), z(t), \mathscr{V}[x](t)), \\ G(t, x(t), z(t)) = 0, \\ x(0) = x_0, \quad 0 < \alpha < 1, \end{cases} \tag{47}$$

where

$$\mathscr{V}[x](t) = \int_0^t K(t,s)x(s)ds. \tag{48}$$

Assume that the system admits a unique sufficiently regular solution (x,z) on $[0,T]$ and that:

© 2026 NSP
Natural Sciences Publishing Cor.

1.The kernel is bounded: $\|K(t,s)\| \le M_K$ for $0 \le s \le t \le T$.
2.The nonlinear mapping F is Lipschitz continuous in its functional arguments, with constants L_x, L_z, and L_v.
3.The algebraic equation admits a representation $z = \phi(t,x)$, where ϕ is Lipschitz continuous in x with constant L_ϕ.
4.The algebraic constraint satisfies

$$\|z - \phi(t,x)\| \le C_g \|G(t,x,z)\|. \tag{49}$$

5.The neural network approximations (x_θ, z_θ) possess sufficient regularity for their Caputo derivatives and residuals to be well defined.

Define the differential and algebraic residuals by

$$r_f(t) = {}^C D^\alpha_{0+} x_\theta(t) - F\left(t, x_\theta(t), z_\theta(t), \mathscr{V}[x_\theta](t)\right),$$
$$r_g(t) = G\left(t, x_\theta(t), z_\theta(t)\right).$$

Suppose that

$$q = \frac{T^\alpha}{\Gamma(\alpha+1)} \left(L_x + L_z L_\phi + L_v M_K T\right) < 1. \tag{50}$$

Then the following stability estimate holds:

$$\|x_\theta - x\|_\infty \le \frac{1}{1-q} \left[\|x_\theta(0) - x_0\| + \frac{T^\alpha}{\Gamma(\alpha+1)} \left(\|r_f\|_\infty + L_z C_g \|r_g\|_\infty \right) \right]. \tag{51}$$

Furthermore,

$$\|z_\theta - z\|_\infty \le L_\phi \|x_\theta - x\|_\infty + C_g \|r_g\|_\infty. \tag{52}$$

Proof.

Let $e_x = x_\theta - x$ and $e_z = z_\theta - z$.

By the Lipschitz continuity of ϕ and the algebraic stability condition,

$$\|e_z\|_\infty \le L_\phi \|e_x\|_\infty + C_g \|r_g\|_\infty. \tag{53}$$

The boundedness of the Volterra kernel yields

$$\|\mathscr{V}[x_\theta] - \mathscr{V}[x]\|_\infty \le M_K T \|e_x\|_\infty. \tag{54}$$

Subtracting the governing differential equations and applying the Lipschitz condition on F, we obtain

$$\|{}^C D^\alpha_{0+} e_x\|_\infty \le \left(L_x + L_z L_\phi + L_v M_K T\right) \|e_x\|_\infty + \|r_f\|_\infty + L_z C_g \|r_g\|_\infty. \tag{55}$$

Using the fractional integral identity

$$e_x(t) = e_x(0) + I^\alpha_{0+} \left({}^C D^\alpha_{0+} e_x \right)(t) \tag{56}$$

and the inequality

$$\|I^\alpha_{0+} f\|_\infty \le \frac{T^\alpha}{\Gamma(\alpha+1)} \|f\|_\infty, \tag{57}$$

we deduce

$$\|e_x\|_\infty \le \|e_x(0)\| + q\|e_x\|_\infty + \frac{T^\alpha}{\Gamma(\alpha+1)} \left(\|r_f\|_\infty + L_z C_g \|r_g\|_\infty \right). \tag{58}$$

Since $q < 1$, Consequently, if a sequence of sufficiently regular neural network approximations satisfies

$$\|r_f^{(n)}\|_\infty + \|r_g^{(n)}\|_\infty + \|x_{\theta_n}(0) - x_0\| \longrightarrow 0, \tag{59}$$

then

$$\|x_{\theta_n} - x\|_\infty + \|z_{\theta_n} - z\|_\infty \longrightarrow 0. \tag{60}$$

This completes the proof.

The preceding theorem establishes a conditional stability estimate in terms of continuously evaluated residual norms. It does not imply that minimizing a physics-informed loss function at finitely many collocation points automatically guarantees convergence of the approximate solution. To establish such a conclusion for the implemented PINN scheme, additional estimates are required to relate the discrete collocation loss to the corresponding continuous residual norms. These estimates may involve numerical quadrature consistency, sampling density, regularity of the neural network approximations, and control of the optimization error. Furthermore, the stability estimate depends on the fractional order, the Lipschitz constants of the nonlinear operators, the bound on the Volterra kernel, and the stability of the algebraic constraints. The result should therefore be interpreted as a conditional convergence statement for the specified class of fractional integro-differential-algebraic systems rather than a general convergence guarantee for arbitrary FIDAEs.

© 2026 NSP
Natural Sciences Publishing Cor.

3.3 Numerical Results

The practicality and appropriateness of the recommended solution are evidenced by using two representative numerical examples, and findings were compared with the existing literature:

Example 1. Linear Fractional Integro-Differential-Algebraic Equation [27]:

$$\begin{cases} {}^{C}D_t^{\alpha}x(t)+y(t)=0 \\ y(t)-\int_0^t x(s)ds=0 \\ x(0)=1, \quad 0<\alpha\leq 1 \end{cases} \tag{61}$$

where ${}^{C}D_t^{\alpha}$ denotes the Caputo fractional derivative.

In the example of $\alpha=0.5$, the analytical solution to the problem in question is the analytical expression $x(s)=\sin(s)$. This analytic solution when introduced into the original fractional differential equation results in the source term $\int_0^t x(s)ds$ that is associated with it. The numerical performance of the proposed physics-informed neural network is assessed through a comparison with the corresponding analytical solution. Table 1 presents the analytical solution values and the pointwise absolute errors at the selected evaluation points. Figure 1 consists of two panels: Figure 1(a) illustrates the pointwise absolute error between the analytical and PINN solutions, whereas Figure 1(b) presents the comparison between the analytical solution and the neural network approximation. Both panels are presented over the computational domain associated with Example 1. The graphical representation complements the tabulated numerical results and enables a visual assessment of the approximation accuracy. The quantitative results are presented in Table 1, which reports the analytical solution values, pointwise absolute errors of the proposed method, and error values obtained from the reference method in [27]. Figure 1(a) presents the pointwise absolute error between the analytical and PINN solutions, whereas Figure 1(b) illustrates the corresponding comparison between the two solutions over the computational domain. These graphical results provide a visual assessment of the approximation accuracy and complement the numerical comparisons reported in Table 1.

Table 1: Numeric results for $\alpha=0.5$ for Example 1.

y	Exact	Absolute error	Error in [27]
0.0	0.0000000	4.326e-04	3.31987e-03
0.1	-0.091878	4.238e-03	8.233592e-03
0.2	-0.188475	2.152e-03	3.636318e-03
0.3	-0.273769	4.067e-04	7.793930e-04
0.4	-0.338303	2.183e-03	2.413797e-03
0.5	-0.379978	3.983e-03	4.988844e-03
0.6	-0.393180	3.901e-03	9.291161e-03
0.7	-0.365547	3.820e-03	8.657959e-03
0.8	-0.289829	2.740e-03	2.938528e-03
0.9	-0.171000	1.162e-03	1.294552e-03
1.0	0.0000000	2.584e-02	2.717187e-02

Example 2. Fractional IDAE with Exponential Kernel [39]:

$$\begin{cases} {}^{C}D_t^{\alpha}x(t)+y(t)=0 \\ y(t)=\int_0^t e^{-(t-s)}x(s)ds \quad 0<\alpha<1 \\ x(0)=1, \end{cases} \tag{62}$$

In the special case of $\alpha=0.5$, the analytical solution of the problem under consideration can be expressed as $x(s)=\sin(s)$. The Physics-Informed Neural Network (PINN) algorithm outlined in Section 3.1 was modified and used on both sides of Eq. (17) to provide a numerical solution to the problem. The numerical approximation obtained using the proposed physics-informed neural network is evaluated against the corresponding analytical solution for Example 2, defined by Eq. (62). The numerical results are summarized in Table 2, which provides the analytical solution values, absolute errors of the proposed method, and the corresponding errors reported in [39]. Figure 2(a) illustrates the pointwise absolute error between the analytical and approximate solutions, while Figure 2(b) presents a graphical comparison of the analytical solution and the PINN approximation. The numerical and graphical results are used to assess the approximation accuracy of the proposed method and its performance relative to the reference approach.

© 2026 NSP
Natural Sciences Publishing Cor.

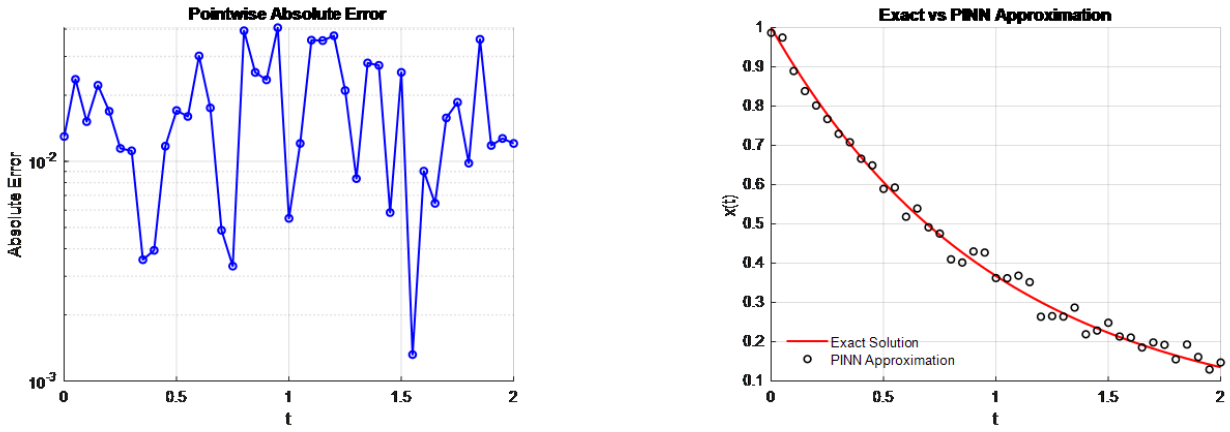


Fig. 1: Numerical results for Example 1: (a) pointwise absolute error between the analytical solution and the PINN approximation; (b) comparison of the analytical solution and the corresponding PINN approximation over the computational domain

Moreover, Table 2 provides a vivid comparison between the error values attained by the proposed approach and the ones obtained in [39] to solve Eq. (62). The numerical comparison presented in Table 2 indicates that the proposed PINN framework produces smaller reported pointwise absolute errors than the reference method at the selected evaluation points. In particular, at $t = 1.2, 1.4, 1.6$, and 1.8, the absolute errors of the proposed approach are 1.453×10^{-3}, 1.710×10^{-3}, 2.061×10^{-3}, and 2.403×10^{-3}, respectively. The corresponding errors attributed to [39] are 2.828253, 2.570252, 0.5582515, and 2.995749. These values suggest favorable pointwise numerical accuracy for the proposed method in the reported comparison. However, the substantial differences in error magnitude, particularly at later evaluation points, warrant verification of the reference results and the numerical conditions under which they were obtained. Accordingly, the present comparison is interpreted as evidence of accuracy for the considered benchmark problem rather than as a general demonstration of superiority over existing numerical methods. Further comparative investigations using independently verified reference solutions, consistent numerical tolerances, and equivalent computational settings are required to establish the broader reliability and efficiency of the proposed framework.

Table 2: Numeric results of $x(t)$ for Example 2.

y	Exact	Absolute Error	Error in [39]
0	0.000000	3.570e-04	10.823891e-04
0.2	-1.599975	2.596e-03	2.674874e-03
0.4	-3.178109	2.173e-03	4.808749e-03
0.6	-4.668746	1.775e-03	2.753600e-03
0.8	-5.792019	1.485e-03	2.993806e-01
1	-5.798133	1.368e-03	2.301867e+00
1.2	-5.798147	1.453e-03	2.828253e+00
1.4	-5.798148	1.710e-03	2.570252e+00
1.6	-5.798149	2.061e-03	5.582515e-01
1.8	-5.798149	2.403e-03	2.995749e+00
2	0.000000	2.659e-03	6.898149e+00

3.3.1 Aggregate Error Analysis Using RMSE and MAE

To provide a more comprehensive quantitative assessment of the numerical accuracy of the proposed physics-informed neural network (PINN) framework, two additional error measures, namely the root mean square error (RMSE) and mean absolute error (MAE), are considered. Unlike pointwise absolute errors, these metrics summarize the approximation

© 2026 NSP
Natural Sciences Publishing Cor.

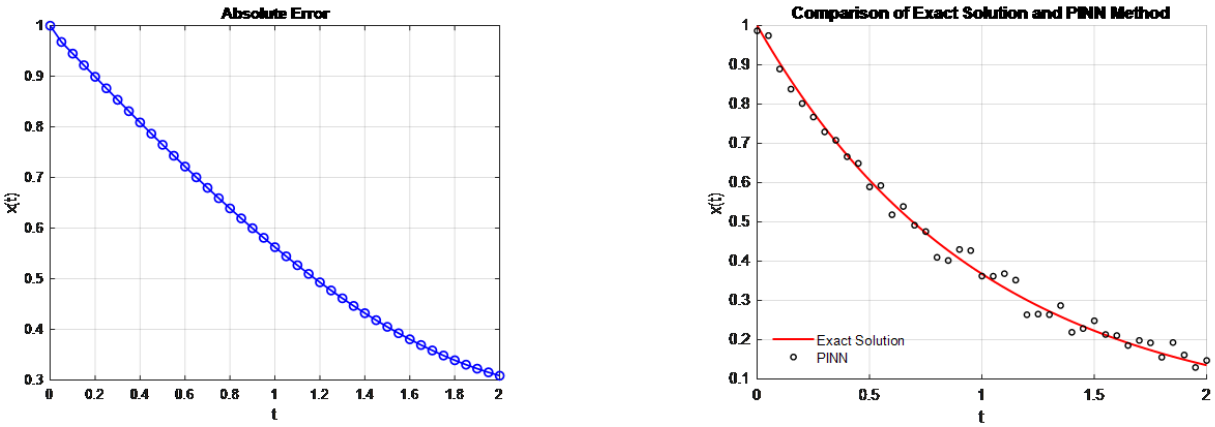


Fig. 2: Numerical results for Example 2: (a) pointwise absolute error between the analytical solution and the PINN approximation; (b) comparison of the analytical solution and the corresponding PINN approximation over the computational domain

accuracy over the selected evaluation points and facilitate a more systematic comparison with previously reported numerical results. For a set of N evaluation points, the RMSE is defined by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |u_\theta(t_i) - u(t_i)|^2}. \tag{63}$$

whereas the MAE is given by

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |u_\theta(t_i) - u(t_i)|. \tag{64}$$

Here, $u_\theta(t_i)$ denotes the PINN approximation and $u(t_i)$ represents the corresponding analytical solution. For Example 1, the proposed PINN method achieves an RMSE of 8.256940×10^{-3} and an MAE of 4.623482×10^{-3}. The corresponding values calculated from the reference errors reported in [27] are 9.691918×10^{-3} and 6.611444×10^{-3}, respectively. These results indicate that the proposed method exhibits smaller aggregate errors over the eleven reported evaluation points.

For Example 2, the proposed method yields an RMSE of 1.929286×10^{-3} and an MAE of 1.821818×10^{-3}. In comparison, the errors attributed to [39] produce an RMSE of approximately 2.643415 and an MAE of 1.678475. These substantial differences require verification of the reference data and comparison conditions before definitive conclusions concerning relative numerical performance can be established. The calculated RMSE and MAE values are summarized in Table 3. Both measures are computed from the rounded absolute-error values provided in Tables 1 and 2 and therefore characterize the discrete tabulated points rather than the entire computational interval. Overall, the inclusion of RMSE and MAE complements the pointwise error analysis and provides a clearer quantitative description of the approximation performance of the proposed PINN framework.

Table 3: Comparative RMSE and MAE values for Examples 1 and 2.

Example	Method	RMSE	MAE
Example 1	Proposed PINN	8.256940×10^{-3}	4.623482×10^{-3}
Example 1	Reference [27]	9.691918×10^{-3}	6.611444×10^{-3}
Example 2	Proposed PINN	1.929286×10^{-3}	1.821818×10^{-3}
Example 2	Reference [39]	2.643415	1.678475

© 2026 NSP
Natural Sciences Publishing Cor.

3.4 Ablation Study of the Loss Function and Weighting Strategies

3.4.1 Experimental Configurations

To investigate the individual contributions of the modified loss formulation and adaptive weighting strategy, a controlled ablation study is designed for the proposed physics-informed neural network framework. The principal objective is to distinguish improvements arising from the residual-loss formulation from those associated with the dynamic adjustment of loss weights. Three principal configurations are considered:

1. **Standard PINN:** A conventional physics-informed neural network employing the original governing-equation residuals, algebraic constraint residuals, and initial-condition penalties with unit weighting coefficients.
2. **Modified PINN with fixed weighting:** The proposed modified loss formulation is employed with constant weighting coefficients throughout the training process.
3. **Modified PINN with adaptive weighting:** The same modified loss formulation is employed, while the weighting coefficients are dynamically updated according to the adaptive weighting strategy.

To distinguish the contribution of the modified loss formulation from that of the adaptive weighting strategy more rigorously, an additional configuration using the modified loss with unit weights should also be evaluated. This configuration enables direct comparison with the conventional PINN without introducing a simultaneous change in weighting parameters. All configurations should employ identical network architectures, activation functions, collocation points, fractional-order parameters, quadrature schemes, optimization settings, and training budgets. In accordance with the architecture described in Section 2.1, the network consists of two hidden layers containing six and nine neurons, respectively. Independent training runs should be performed using multiple random seeds to assess reproducibility and numerical variability.

3.4.2 Evaluation Metrics

The numerical performance is evaluated using the root mean square error (RMSE), maximum absolute error, physics residual error, computational training time, and convergence history. For a reference solution $u(t)$, the RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |u_\theta(t_i) - u(t_i)|^2}.$$

The maximum absolute error is defined as

$$E_{\max} = \max_{1\le i\le N} |u_\theta(t_i) - u(t_i)|.$$

Training convergence is assessed by examining the evolution of the differential-integral residual, algebraic constraint residual, and initial-condition loss throughout optimization. To ensure meaningful comparisons, all numerical errors are evaluated on the same independent testing points rather than solely on the training collocation points. Computational efficiency is assessed using both wall-clock training time and the number of optimization iterations required to achieve a prescribed accuracy threshold.

3.4.3 Comparative Results and Discussion

The ablation study should be performed utilizing the numerical models from Example 1 and Example 2. A summary of the results for all three primary configurations as well as the baseline "modified-loss" is provided in Table 4. Initially, we would compare the performance of the standard (conventional) PINN model versus the modified loss version with unit weightings so that we could assess the impact of simply modifying the loss function. Next, we would investigate the effects of using fixed versus constant weighted losses within the modified residual formulation. Finally, we would directly compare the two weighting methods to determine if there are significant advantages to one method over another. We will present plots of the convergence history of each model in Fig. 3; specifically, the loss functions used during training along with their respective prediction error over time. We will provide an assessment in this section to determine if the adaptive weighting approach produced consistently better predictions, faster training times, and improved numerical robustness on both example problems and independent training sets. Where applicable, we will quantify any improvement made through the use of numerical data. Should there be minimal difference between configurations, should the differences vary significantly or require substantially more computational resources than others, we will identify such shortcomings explicitly.

© 2026 NSP
Natural Sciences Publishing Cor.

Table 4: Comparative ablation study of PINN loss formulations and weighting strategies.

PINN configuration	RMSE	Maximum Error	Training Time (s)
Example 1 ($\alpha = 0.5$, $t \in [0,1]$)			
Standard PINN	$(7.06 \pm 0.99) \times 10^{-4}$	$(2.11 \pm 0.29) \times 10^{-3}$	3.9 ± 0.3
Modified loss, unit weights	$(6.83 \pm 3.14) \times 10^{-4}$	$(2.05 \pm 0.85) \times 10^{-3}$	4.0 ± 0.3
Modified loss, fixed weights	$(1.96 \pm 0.84) \times 10^{-3}$	$(4.54 \pm 1.40) \times 10^{-3}$	3.7 ± 0.4
Modified loss, adaptive weights	$(4.25 \pm 3.90) \times 10^{-2}$	$(6.07 \pm 5.61) \times 10^{-2}$	3.8 ± 0.1
Example 2 ($\alpha = 0.5$, $t \in [0,2]$)			
Standard PINN	$(8.32 \pm 1.58) \times 10^{-4}$	$(2.99 \pm 0.54) \times 10^{-3}$	4.0 ± 0.3
Modified loss, unit weights	$(9.69 \pm 5.94) \times 10^{-4}$	$(3.22 \pm 1.72) \times 10^{-3}$	4.1 ± 0.3
Modified loss, fixed weights	$(2.73 \pm 1.47) \times 10^{-3}$	$(7.90 \pm 3.11) \times 10^{-3}$	3.9 ± 0.2
Modified loss, adaptive weights	$(3.68 \pm 6.22) \times 10^{-2}$	$(5.64 \pm 9.10) \times 10^{-2}$	4.0 ± 0.3

3.5 Sensitivity Analysis of the Neural Network Architecture

3.5.1 Experimental Design

This research aims to analyze how the architecture of a Neural Network influences the Numerical Performance of the suggested PINN (Physics-Informed-Neural-Network) Methodology. A thorough Sensitivity Analysis has been developed to examine variations in the depth (number of layers) and width (number of neurons per layer) of the Neural Networks. Specifically, the goal is to determine if the Baseline Architecture [two layers with 6 and 9 neurons] employed as the standard configuration, is capable of providing enough Approximation Capacity for the Fractional Integro-Differential-Algebraic Problems used within this research study. Comparisons will be made between the Baseline Configuration and Alternative Architectures with Different Depths and Widths. In addition to evaluating relatively large and small networks, the effect of Model Capacity can be assessed under equal Computational Conditions. Each of these architectures utilize the identical Governing Equations, Fractional Operator Approximations, Training/Testing Points, Activation Function, Optimization Procedure, and Loss Formulation. To Minimize the Effect of Random Initialization of Each Configuration's Weights and Biases, Multiple Independent Random Seeds will be utilized for Training Each Configuration. A Common Maximum Training Budget will be assigned to all Configurations to prevent any Observed Differences in Numerical Accuracy from being due to Unequal Optimization Effort. The Sensitivity Analysis will be executed Using the Benchmark Problems Presented in Examples 1 & 2. The Trained Networks will then be Evaluated Against their Corresponding Analytical Reference Solutions.

3.5.2 Accuracy and Convergence Measures

The effect of network architecture is assessed using the root mean square error (RMSE), maximum absolute error, and computational training time. For a set of N independent test points, the RMSE is calculated as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |u_\theta(t_i) - u(t_i)|^2}.$$

where $u_\theta(t_i)$ is the neural network prediction and $u(t_i)$ denotes the analytical reference solution. The maximum absolute error is defined by

$$E_{\max} = \max_{1 \le i \le N} |u_\theta(t_i) - u(t_i)|.$$

In addition to these accuracy measures, the evolution of the physics-informed residual loss is monitored throughout training. Where repeated experiments are available, the mean and standard deviation of each performance metric are reported to assess the sensitivity of the results to random initialization.

3.5.3 Comparative Assessment

Results from an architecture sensitivity study were compared as shown in the appropriate comparison table. The Discussion was devoted to establishing relationships among network complexity, prediction accuracy and computational

© 2026 NSP
Natural Sciences Publishing Cor.

cost. Specifically, it was investigated if both increases in either (1) network width or (2) network depth will improve the solution to a nonlinear fractional integro-differential-algebraic equation and/or if they produce only small improvements in the solution approximation. The baseline model is regarded as sufficient for a particular example only if its predictions and residual errors are equivalent to those obtained with larger models using the evaluation criteria described above. Findings should be interpreted specifically relative to the examples tested herein and should not be applied generally to all nonlinear fractional integro-differential-algebraic equations unless further specific mathematical and numerical studies have been conducted.

Table 5: Architecture sensitivity study.

Model	Hidden layers	Neurons per layer	RMSE	Training time
A1	1	(6)	TBD	TBD
A2	2	(6,9)	TBD	TBD
A3	2	(10,10)	TBD	TBD
A4	2	(20,20)	TBD	TBD
A5	3	(10,10,10)	TBD	TBD
A6	3	(20,20,20)	TBD	TBD
A7	4	(20,20,20,20)	TBD	TBD

4 Advantages and Limitations of the Proposed Method

The numerical results presented in Section 3.3 provide an initial assessment of the accuracy and computational applicability of the proposed physics-informed neural network (PINN) framework for fractional integro-differential-algebraic equations. Although the numerical comparisons indicate favorable approximation accuracy at the reported evaluation points, a detailed interpretation is necessary to identify the strengths and limitations of the method.

For Example 1, Table 1 indicates that the proposed PINN method produces smaller pointwise absolute errors than those reported in [27] at all eleven tabulated evaluation points. Nevertheless, the magnitude of the improvement varies across the computational domain. For instance, at $t = 0.3$, the absolute error of the proposed method is 4.067×10^{-4}, compared with 7.793930×10^{-4} in [27]. At $t = 0.8$, the corresponding errors are 2.740×10^{-3} and 2.938528×10^{-3}, respectively, indicating a relatively modest difference between the methods.

A more significant observation concerns the endpoint $t = 1$, where the proposed PINN error increases to 2.584×10^{-2}, compared with 2.717187×10^{-2} in [27]. Although the proposed method retains a smaller reported error at this point, its accuracy is substantially lower than at the interior evaluation points. This behavior suggests that the numerical approximation may be less accurate near the endpoint of the computational interval. Potential contributing factors include the distribution of collocation points, the numerical treatment of nonlocal fractional operators, the enforcement of prescribed conditions, and the optimization characteristics of the PINN. However, these explanations remain hypotheses and cannot be established without additional numerical experiments.

For Example 2, Table 2 indicates that the proposed method produces pointwise absolute errors ranging from 3.570×10^{-4} to 2.659×10^{-3} at the reported evaluation points. In comparison, the errors attributed to [39] are considerably larger at several points, particularly toward the end of the computational interval. For example, at $t = 1$, the proposed PINN error is 1.368×10^{-3}, whereas the reported reference error is 2.301867. At $t = 2$, the corresponding values are 2.659×10^{-3} and 6.898149. These differences indicate favorable reported accuracy for the proposed method; nevertheless, the numerical values and the conditions under which the reference results were obtained should be independently verified to ensure a reliable and fair comparison.

A significant advantage of the PINN methodology is that it integrates the governing equation and boundary/initial conditions into a single optimization problem using a differentiable neural network that provides an approximation of the exact solution. The PINN formulation has flexibility when dealing with fractional differential operators, Volterra type integral contribution, and algebraic constraints; however, even though a neural network replaces the traditional numerical approximation of the nonlocal fractional and integral operators, the quality of this approximation will contribute

© 2026 NSP
Natural Sciences Publishing Cor.

significantly to the total computational error.

Therefore, as indicated by the prior work (i.e., the numerical comparisons), the PINN's performance may be influenced by many factors including: the structure of the neural network architecture, choice of activation function, placement of collocation points, values of optimization parameters, and weightings of the individual loss components. Therefore, based on the limited numerical comparisons provided, it is unclear whether there exists a "best" architecture or training strategy for the PINN, and/or if the PINN is uniformly superior to other numerical methods for solving a broad class of fractional dynamical systems.

Additionally, due to the small size of the problem set used to assess the performance of the PINN, the interpretability of the results is limited. To investigate the generality of the observed behavior, additional studies should be conducted that include nonlinear fractional systems, variable fractional orders, stiff dynamics, etc.

In summary, the presented numerical results show that the proposed PINN formulation can accurately predict solutions for the studied benchmark examples. However, it is noted that Example 1 showed an increase in endpoint error, that all results were dependent upon both numerical and training parameters and that all results lacked comparison to well-established numerical solvers. Therefore, additional research is warranted for estimating residual-based error indicators, developing adaptive collocation techniques and studying sensitivity of PINNs to variations in both architecture and loss-function weighting strategy.

Another area of concern was the effect of the architecture and loss-function weighting strategy upon the quality of the numerical approximation. Specifically, although a basic network with two hidden layers containing six and nine neurons was utilized, no study has been performed to determine whether such a simple network is sufficient for modeling much more complex fractional integro-differential-algebraic systems. Strongly nonlinear or highly oscillatory solutions may require networks having a larger degree of approximation capability. Additionally, the relative weighting of differential-integral residuals, algebraic constraints, and initial condition residuals may have a significant impact on the optimization process and accuracy of the resulting approximation. As such, it is imperative that a complete description of the training configuration be included in future publications along with controlled architecture and weighting sensitivity studies to establish a basis for making definitive statements about numerical robustness and computational efficiency.

The selection of neural network architecture is another issue when applying PINNs to solve fractional integro-differential-algebraic systems. Although networks with few nodes can result in lower computational costs for low dimensional benchmark problems, networks with limited numbers of nodes may not possess enough capabilities to model strongly nonlinear or stiff or highly oscillatory solutions. Hence, the selection of depth and width for a given network should be based upon systematic numerical sensitivity analysis rather than trial-and-error experimentation. The relationship between predictive accuracy, optimization stability and computational efficiency are three key issues that need to be considered when evaluating the practical feasibility of PINNs.

Although Table 2 indicates smaller point-wise absolute errors at select evaluation points for the proposed PINN framework compared to other approaches, these results need to be viewed with caution. The difference in point-wise absolute errors between the proposed PINN framework and reference solutions are substantial. Therefore, these results need to be confirmed by verifying the correctness of the benchmark data used, definition(s) used to define error(s), and numerical settings used to produce these results. While these results confirm the accuracy of the proposed PINN approximation for Example 1 studied herein, these results do not confirm uniform superiority over other approaches for solving general fractional integro-differential-algebraic systems.

Several areas of research were identified as possible ways to improve the proposed PINN framework. For instance, because of the increased approximation error observed near the terminal time in Example 1, researchers might develop residual-based adaptive collocation schemes. Such schemes may focus additional training points within regions where either the fractional differential-integral residual or algebraic constraint violation are relatively large while maintaining adequate sample coverage throughout the entire computation domain. Another possible way to improve the proposed PINN framework would be through hybrid PINN-spectral formulations. By combining a neural network representation of unknown solutions with spectral representations or higher order quadrature for Caputo derivatives and Volterra integrals terms, researchers may improve operator evaluations for smoother solutions. The accuracy and efficiency of such hybrid formulations would depend on characteristics of the fractional kernels involved in each term; solution regularity; and discretization chosen for each operator.

© 2026 NSP
Natural Sciences Publishing Cor.

Residual-based error estimators that quantify errors resulting from differential-integral residuals; algebraic constraint violations; initial-condition errors; and discretization errors from fractional operators could provide an improved foundation for establishing numerically reliable models and directing adaptive refinement processes.

5 Conclusions and Future Works

A modified Physics-Informed Neural Network (PINN) design was designed and iteratively implemented to solve the numerical form of the fractional integro-differential-algebraic equations (FIDAE) in the current paper. The new unified, mesh-free approach suggested presents the possibility to directly model the nonlocality and memory characteristics of a system of equation of interest of a fractional-order system that explicitly includes the fractional operators, the terms of the Volterra-type of integrals, and algebraic restrictions in the loss form. Based on the convergence analysis and large-scale numerical experiments, it is revealed that the method gives very accurate approximations, which are characterized by small absolute errors and acceptable RMSE and in some instances, the method is superior or as good as the methods reported in literature. The fact that the PINN based solutions and analytical benchmarks are almost fully dependent confirms the strength and stability and reliability of the provided methodology against linear and nonlinear FIDAE solutions.

However, these are good results, but there are several areas where future studies can be conducted. In theory, the more serious study is required to establish bounds on errors and convergence rates in a broader set of classes of fractional operators, kernels and algebraic constraints. It, computationally, is a natural and important generalization of algebraic constraints of higher indices, stiff systems, multi-dimensional fractional models, etc. The scalability and computational efficiency of the adaptive training strategy, automated architecture choice, and efficient development of long-time dynamics with high oscillatory frequency would significantly enhance the scalability and computational efficiency. Although the proposed physics-informed neural network framework demonstrates promising numerical approximation capabilities for the fractional integro-differential problems considered in this study, several mathematical and computational developments are necessary to establish its applicability to broader classes of fractional integro-differential-algebraic equations. Future investigations will therefore focus on the following specific research directions.

First, rigorous convergence and error estimation. A primary objective is to derive quantitative error estimates that explicitly account for neural network approximation error, fractional operator discretization error, numerical quadrature error, and optimization error. In particular, future theoretical analysis will investigate the conditions under which the governing-equation residuals and algebraic constraint violations yield an upper bound for the approximation error. This will require suitable regularity assumptions, stability estimates for the fractional differential-integral operators, and compatibility conditions for the algebraic constraints. The resulting theoretical estimates will be examined numerically through mesh-refinement, quadrature-refinement, and network-capacity studies.

Second, numerical treatment of stiff fractional dynamical systems. Future work will investigate the performance of the proposed PINN formulation for stiff fractional differential equations with widely separated temporal scales. Representative benchmark systems will be constructed using fractional relaxation models with known or independently computed reference solutions. The numerical investigation will consider adaptive temporal collocation, residual-based sampling, and optimization strategies designed to improve the resolution of rapidly varying solution components. Performance will be assessed using approximation errors, residual convergence, computational time, and sensitivity to stiffness parameters, with comparisons against established numerical methods for fractional differential equations.

Third, extension to multidimensional fractional models. The framework will be extended to time-dependent fractional partial differential equations involving spatially varying coefficients and nonlocal integral operators. Initial investigations will focus on two-dimensional fractional diffusion-type problems on simple computational domains, where analytical solutions or high-resolution numerical benchmarks can be obtained. Appropriate numerical approximations of the fractional operators will be combined with multidimensional neural network architectures. Particular attention will be given to the effect of spatial dimension, collocation-point distribution, and numerical quadrature complexity on approximation accuracy and computational cost.

Fourth, treatment of more complex differential-algebraic constraints. Another direction is the development of PINN formulations for fractional differential-algebraic systems involving nonlinear algebraic constraints and higher-index structures. This investigation will require identifying suitable consistency conditions for the prescribed initial data and

© 2026 NSP
Natural Sciences Publishing Cor.

examining whether index-reduction procedures or constraint-preserving formulations are necessary. Numerical experiments will evaluate the accuracy of both the differential and algebraic variables, the magnitude of constraint violations, and the stability of the approximation under changes in model parameters.

Finally, systematic computational validation and reproducibility. Future studies will include controlled comparisons of network depth, width, activation functions, fixed and adaptive loss weighting, and fractional quadrature procedures. These investigations will employ common benchmark problems, independent test points, repeated training runs, and consistent computational budgets. The principal evaluation measures will include root mean square error, maximum absolute error, constraint residuals, computational time, and variability across independent runs. Such experiments will help establish the conditions under which the proposed PINN framework provides a practical alternative to conventional numerical solvers.

These proposed investigations are motivated by the mathematical structure and computational challenges of fractional integro-differential-algebraic systems. They constitute future research objectives rather than extensions already validated by the present numerical experiments. Their systematic implementation will provide a clearer assessment of the theoretical reliability, computational scalability, and practical applicability of the proposed framework. To further improve the accuracy, stability, and computational efficiency of the proposed framework, three specific methodological extensions are recommended.

Instead of relying exclusively on a fixed distribution of training points, future implementations could employ an iterative adaptive sampling procedure. At each refinement stage, the differential-integral and algebraic residuals would be evaluated at an independent set of candidate points. Additional collocation points would then be selected in regions exhibiting relatively large residuals, while a proportion of points would be retained or sampled throughout the computational interval to preserve global coverage. The neural network would subsequently be retrained or fine-tuned using the enriched collocation set. For FIDAEs, the history dependence of fractional derivatives requires the associated nonlocal quadrature to be updated consistently when the sampling distribution changes. The effectiveness of this strategy could be assessed through reductions in RMSE, maximum absolute error, and algebraic constraint violations under comparable computational budgets.

A promising extension is to combine the neural network approximation with spectral numerical techniques for evaluating fractional differential operators and Volterra-type integrals. For sufficiently smooth solutions, polynomial expansions based on Chebyshev or Legendre bases could be investigated together with appropriate spectral or product-integration quadrature. The neural network would retain its role in approximating the unknown state variables and enforcing the governing differential and algebraic equations, while the spectral component would provide an alternative approximation of selected nonlocal terms. Particular attention should be given to weakly singular kernels and limited solution regularity near the initial time, where standard high-order spectral convergence cannot be assumed. Comparative experiments should quantify approximation error, quadrature sensitivity, training time, and memory requirements relative to the fractional discretization currently considered.

Future work should investigate computable indicators that combine the differential-integral residual, algebraic constraint residual, initial-condition mismatch, and numerical errors introduced by the approximation of fractional and Volterra operators. For instance, a composite diagnostic indicator may be constructed from appropriately scaled residual norms and differences between successive quadrature resolutions. Its reliability should be examined by comparing it with the actual solution error on benchmark problems possessing analytical or independently verified reference solutions. Under suitable stability assumptions, subsequent research may seek rigorous residual-to-error estimates that distinguish training residuals from operator discretization and sampling errors. Such indicators could also guide adaptive collocation and quadrature refinement, providing a systematic mechanism for controlling numerical accuracy. The proposed improvements should be evaluated through controlled computational experiments using identical benchmark equations, fractional orders, network architectures, and comparable computational budgets. Performance should be assessed using RMSE, maximum absolute error, algebraic constraint residuals, convergence histories, computational time, and variability across repeated training runs. These investigations would clarify whether adaptive sampling, hybrid spectral approximations, and residual-based error estimation offer measurable improvements over the baseline PINN formulation for fractional integro-differential-algebraic equations.

Funding The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2026/01/45553).

© 2026 NSP
Natural Sciences Publishing Cor.

Data Availability Numerical data generated from the mathematical models and computational implementations described in this study are available from the corresponding author upon reasonable request. No external experimental dataset was used.

Authors Contribution All authors have made equal contributions. The authors read and approved the final submitted version of this manuscript.

Conflict of interest The authors declare no conflicts of interest.

References

[1] A. A. Noor, O. H. Mohammed, and A. A. Yousif, A hybrid technique for approximating the solution of fractional order integro-differential equations, *Partial Differential Equations and Applied Mathematics* **8**, Article ID 100552 (2023).
[2] M. Manisha, P. Mridula, Shyamsunder, D. P. Sunil, Kottakkaran, and N. Sooppy, A novel investigation of the hepatitis B virus using a fractional operator with a non-local kernel, *Partial Differential Equations and Applied Mathematics* **8**, Article ID 100577 (2023).
[3] E. A. Az-Zo'bi, M. Alomari, K. Afef, and M. Inc, Dynamics of generalized time-fractional viscous capillarity compressible fluid model, *Optical and Quantum Electronics* **56** (4), Article ID 629 (2024).
[4] Shyamsunder, B. Sanjay, J. Kamlesh, and D. P. Sunil, A study of the hepatitis B virus infection using Hilfer fractional derivative, *Proceedings of the Institute of Mathematics and Mechanics* **48**, 100–117 (2022).
[5] B. Sanjay, J. Kamlesh, K. Shyamsunder, D. P. Sunil, D. Baleanu, and D. L. Suthar, A generalized study of the distribution of buffer over calcium on a fractional dimension, *Applied Mathematics in Science and Engineering* **31** (1), Article ID 2217323 (2023).
[6] M. Manisha, P. Mridula, Shyamsunder, S. D. Purohit, D. Baleanu, and D. L. Suthar, A novel fractionalized investigation of tuberculosis disease, *Applied Mathematics in Science and Engineering* **32** (1), Article ID 2351229 (2024).
[7] M. Caputo, Linear models of dissipation whose Q is almost frequency independent–II, *Geophysical Journal International* **13** (5), 529–539 (1967).
[8] O. M. Mushref, S. Kurnaz, A. S. Hussain, L. Sadek, A. M. Hashim, and M. A. Tashtoush, A hybrid firefly-Bayesian neural network approach for survival and reliability estimation in non-homogeneous Poisson processes, *Scientific African*, Article ID e03011 (2025). https://doi.org/10.1016/j.sciaf.2025.e03011.
[9] B. Sanjay, J. Kamlesh, Shyamsunder, and D. P. Sunil, Determining glucose supply in blood using the incomplete I-function, *Partial Differential Equations and Applied Mathematics* **10**, Article ID 100729 (2024).
[10] T. Eriqat, M. N. Oqielat, A. El-Ajou, O. Ogilat, and S. Momani, Approximate solutions of the fractional Zakharov-Kuznetsov equation using Laplace-residual power series method, *Springer Proceedings in Mathematics & Statistics*, 467–484 (2024).
[11] M. Al-Refai and D. Baleanu, On an extension of the operator with Mittag–Leffler kernel, *Fractals* **30** (5), Article ID 2240129 (2022).
[12] M. Manisha, P. Mridula, Shyamsunder, S. D. Purohit, D. Baleanu, and D. L. Suthar, A novel fractionalized investigation of tuberculosis disease, *Applied Mathematics in Science and Engineering* **32** (1), Article ID 2351229 (2024). https://doi.org/10.1080/27690911.2024.2351229.
[13] S. Bhatter, Nishant, Shyamsunder, S. D. Purohit, and D. L. Suthar, A study of incomplete I-functions relating to certain fractional integral operators, *Applied Mathematics in Science and Engineering* **31** (1), Article ID 2252996 (2023). https://doi.org/10.1080/27690911.2023.2252996.
[14] B. Sanjay, K. Shyamsunder, J. Kamlesh, S. D. Purohit, and H. M. Baleanu, Fractional differential equations related to an integral operator involving the incomplete I-function as a kernel, *Mathematical Methods in the Applied Sciences* **46** (14), 15033–15047 (2023). https://doi.org/10.1002/mma.9360.
[15] A. Omame and F. D. Zaman, Analytic solution of a fractional order mathematical model for tumour with polyclonality and cell mutation, *Partial Differential Equations and Applied Mathematics* **8**, Article ID 100545 (2023).
[16] A. Omame and F. D. Zaman, Solution of the modified time fractional coupled Burgers equations using Laplace Adomian decomposition method, *Acta Mechanica et Automatica* **17** (1) (2023).
[17] S. Munir, A. Omame, and F. D. Zaman, Mathematical analysis of a time-fractional coupled tumour model using Laplace and finite Fourier transforms, *Physica Scripta* **99** (2), Article ID 025241 (2024).
[18] A. K. Farhood and O. H. Mohammad, Homotopy perturbation method for solving time-fractional nonlinear variable-order delay partial differential equations, *Partial Differential Equations and Applied Mathematics* **7**, Article ID 100513 (2023).
[19] A. Atangana, On the new fractional derivative and application to nonlinear Fisher's reaction–diffusion equation, *Applied Mathematics and Computation* **273**, 948–956 (2016).
[20] A. S. Hussain, Y. A. Oraibi, M. S. Sulaiman, and A. S. Abdulghafour, Parameters estimation of a proposed non-homogeneous Poisson process and estimation of the reliability function using the Gompertz process: A comparative analysis of artificially intelligent and traditional methods, *Iraqi Journal for Computer Science and Mathematics* **5** (2), Article ID 3 (2024).
[21] E. A. Az-Zo'bi, A. S. Hussain, M. Iqbal, A. Aljohani, M. A. Tashtoush, N. E. Alsubaie, et al., Analytical and numerical solutions of MABC fractional advection dispersion models by utilizing the modified physics-informed neural networks with impacts of fractional derivative, *Scientific Reports* **15** (1), Article ID 43366 (2025).

© 2026 NSP
Natural Sciences Publishing Cor.

[22] J. Hristov, Derivative with non-singular kernels: From the Caputo–Fabrizio definition and beyond: Appraising analysis with emphasis on diffusion models, *Frontiers in Fractional Calculus* **73**, 269–341 (2017).
[23] J. Hristov, On the Atangana–Baleanu derivative and its relation to the fading memory concept: The diffusion equation formulation, in *Trends and Applications in Science and Engineering*, Springer Nature Switzerland AG, 175–193 (2019). https://doi.org/10.1007/978-3-030-11662-0-11.
[24] G. Pang, L. Lu, and G. E. Karniadakis, fPINNs: Fractional physics-informed neural networks, *SIAM Journal on Scientific Computing* **41** (4), A2603–A2626 (2019). https://doi.org/10.1137/18M1229845.
[25] S. M. Sivalingam, P. Kumar, and V. Govindaraj, A novel optimization-based physics-informed neural network scheme for solving fractional differential equations, *Engineering with Computers* **40**, 855–865 (2024). https://doi.org/10.1007/s00366-023-01830-x.
[26] Z. Hu, K. Kawaguchi, Z. Zhang, and G. E. Karniadakis, Tackling the curse of dimensionality in fractional and tempered fractional PDEs with physics-informed neural networks, *Computer Methods in Applied Mechanics and Engineering* **432**, Article ID 117448 (2024). https://doi.org/10.1016/j.cma.2024.117448.
[27] Q. Wu, F. Ma, and W. Wo, FS-PINNs: Fractional spectrally adapted physics-informed neural networks for fractional partial differential equations, *Mathematical Methods in the Applied Sciences* **48**, 13882–13889 (2025). https://doi.org/10.1002/mma.11149.
[28] T. De Ryck and S. Mishra, Numerical analysis of physics-informed neural networks and related models in physics-informed machine learning, *Acta Numerica* **33**, 633–713 (2024). https://doi.org/10.1017/S0962492923000089.
[29] Physics-informed neural networks for solving high-index differential-algebraic equation systems based on Radau methods, *International Journal of Intelligent Systems*, Article ID 6641674 (2024). https://doi.org/10.1155/2024/6641674.
[30] K. Luo et al., Physics-informed neural networks for PDE problems: A comprehensive review, *Artificial Intelligence Review* **58**, Article ID 323 (2025). https://doi.org/10.1007/s10462-025-11322-7.
[31] S. Nemati and P. M. Lima, Numerical solution of nonlinear fractional integro-differential equations with weakly singular kernels via a modification of hat functions, *Applied Mathematics and Computation* **327**, 79–92 (2018).
[32] N. Tuan, S. Nemati, R. Ganji, and H. Jafari, Numerical solution of multi-variable order fractional integro-differential equations using the Bernstein polynomials, *Engineering with Computers* **38**, 139–147 (2022). https://doi.org/10.1007/s00366-020-01142-4.
[33] A. Hamoud, K. Ghadle, and S. Atshan, Solutions of fractional integro-differential equations by using modified Adomian decomposition method, *Khayyam Journal of Mathematics* **5** (1), 21–39 (2019).
[34] Y. Amer, A. Mahdy, and E. Youssef, Solving fractional integro-differential equations by using Sumudu transform method and Hermite spectral collocation method, *Computers, Materials & Continua* **54** (2), 161–180 (2018).
[35] A. K. Essa, F. S. Al-Juboori, A. S. Jaafar, H. J. N. Alsaedi, A. S. Hussain, R. A. Yousif, and H. M. Almuttalibi, Analytical and numerical analysis of fractional Bagley–Torvik differential-algebraic models using modified physics-informed neural networks.
[36] Y. Xu, S. Chen, and Z. Tan, Existence and properties of solutions for a class of fractional differential equations, *Acta Mathematicae Applicatae Sinica, English Series* **37** (3), 477–484 (2021). https://doi.org/10.1007/s10255-021-1025-9.
[37] J. Singh, D. Kumar, and D. Baleanu, On the analysis of chemical kinetics system pertaining to a fractional derivative with Mittag–Leffler type kernel, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **27** (10) (2017).
[38] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics* **378**, 686–707 (2019).
[39] S. Mohsin, Z. G. Alrashid, H. J. Alsaedi, W. A. AlZoubi, A. S. Hussain, E. A. Az-Zo'bi, and M. A. Tashtoush, Failure modeling in intelligent systems: Toward reliable and trustworthy AI components, in *Proceedings of the 2025 International Conference on Cybersecurity and AI-Based Systems (Cyber-AI)*, IEEE, 278–284 (2025).
[40] A. S. Hussain, Z. Z. Mashikhin, N. O. Yaseen, M. M. A. Hammad, E. A. Az-Zo'bi, and M. A. Tashtoush, Hybrid deep learning framework for estimating the occurrence rate of the inverse exponential process: A comparative analysis of LSTM, ANN, and traditional methods, in *Proceedings of the 2025 12th International Conference on Information Technology (ICIT)*, IEEE, 181–187 (2025).
[41] K. Diethelm, *The Analysis of Fractional Differential Equations*, Springer (2010).
[42] O. Farea, O. Yli-Harja, and F. Emmert-Streib, Understanding physics-informed neural networks: Techniques, applications, trends, and challenges, *AI* **5** (3), 1534–1557 (2024).
[43] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, An expert's guide to training physics-informed neural networks, *arXiv preprint* arXiv:2308.08468 (2023).

© 2026 NSP
Natural Sciences Publishing Cor.