AIMS Mathematics, 11(8): 25705–25743.
DOI: 10.3934/math.20261030
Received: 27 February 2026
Revised: 24 April 2026
Accepted: 12 May 2026
Published: 19 August 2026

https://www.aimspress.com/journal/Math

Research article

A comparative study of the fractional differential equations using physics-informed neural networks

Sneha Agarwal and Lakshmi Narayan Mishra*

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore 632 014, Tamil Nadu, India

* **Correspondence:** Email: lakshminarayanmishra04@gmail.com, lakshminarayan.mishra@vit.ac.in; Tel: +919838375431.

Abstract: In this work, we demonstrated the use of neural network techniques for solving fractional differential equations (FDEs) governed by the Caputo–Katugampola derivative, specifically standard neural networks and physics-informed neural networks. Our numerical approach was based on the Volterra integral representation of the generalized fractional operator, which provides a natural way to incorporate memory and non-locality effects into the solution procedure. We designed and studied two learning frameworks: (i) The proposed NFDEs model that approximates the system's dynamics by data-driven learning of the solution trajectory and (ii) the proposed PINFDEs model that informs the learning process by directly embedding the residual form of the governing FDE into the neural network training objective. For numerical implementation, the Caputo–Katugampola derivative was discretized by using a Volterra-type quadrature scheme that accurately and efficiently computed the fractional operator throughout the considered time domain. To demonstrate the performance of the proposed frameworks, four numerical experiments were performed for different fractional orders. Our results emphasized that, while the neural fractional differential equation model can capture the general solution, the proposed PINFDEs model exhibits more accuracy, stability, and generalization for a wide variety of cases, particularly for fractional orders $\gamma < 1$, where the nonlocal effects dominate. The improvement in the performance reflected the fact that the explicit incorporation of the physical constraints during the learning stage significantly reduces overfitting and enhances the robustness of the solution. Moreover, the comparative study confirms that embedding physical knowledge into data-driven neural models substantially enhances their predictive capabilities regarding fractional systems.

Keywords: Caputo–Katugampola fractional derivative; Volterra integral equations; neural networks; physics-informed neural networks
Mathematics Subject Classification: 26A33, 34K37, 45D05, 68T07

Nomenclature

Abbreviations

NFDEs	Neural fractional differential equation(s).
PINFDEs	Physics-informed neural fractional differential equation(s).
C–K	Caputo–Katugampola.
MSE	Mean squared error.
MAE	Mean absolute error.
SSE	Sum of squared error.
R^2	Coefficient of determination.
NSE	Nash-Sutcliffe efficiency.
NN	Neural network(s).
PINNs	Physics-informed neural network(s).
HL	Hidden layer.
P-inf.	Physics-informed.
A/F	Activation function.
LF	Loss function.
FDEs	Fractional differential equation(s).

Operators

${}_cD_u^{\gamma}$	Caputo fractional derivative of order γ.
${}_cI_u^{-\gamma}$	Riemann–Liouville fractional integral of order γ.
${}^{CK}D_t^{\gamma,\beta}$	Caputo–Katugampola fractional derivative of order γ and tunable parameter β.

Roman symbols

$\mathbb{R}$	Set of all real numbers.
λ_1	Weighting parameter for data loss term.
λ_2	Weighting parameter for PINN loss term.

1. Introduction

Fractional calculus originally appeared in 1695, during a correspondence between the German mathematician Gottfried Wilhelm Leibniz and the French mathematician Guillaume François Antoine Marquis de L'Hôpital. The possibility that the order of a derivative could be a real number instead of an integer was raised by L'Hôpital in 1695. The initial studies of fractional calculus were primarily theoretical in nature. Since the 1960s, it has garnered significant interest in the scientific and engineering fields and is utilized extensively in many research fields. Riemann–Liouville, Grünwald-Letnikov [1], Riesz, Weyl [2], and the Caputo [3] representation are some of the definitions of the fractional-order derivative [4]. In economics, scholars have focused a lot of emphasis on the application of fractional calculus in economic models. Neural ODEs and PINNs were successfully integrated to develop P-inf. neural ordinary differential equations (PINODEs). Through the integration of physical principles into NNs, PINODEs improve the precision and comprehensibility of complex dynamical system modeling. A method for learning and solving nonlinear partial differential equations (PDEs)

using Gaussian processes was proposed by Chen et al. [5] to better understand complex dynamical systems. Numerous methods are available for the analytical solution of nonlinear differential equations, mainly nonlinear FDEs in terms of the Caputo derivative [6]. Analytical solutions solve nonlinear FDEs, such as Laplace [7] and finite Hankel transforms [8], and one can use the transform technique, the new iterative Elzaki transform (NIET) method, integral transforms [9], and the modified reduced differential transform (MRDT) method. For the purpose of solving linear and nonlinear mathematical physics equations, the variational iteration method (VIM) and the Adomian decomposition method (ADM) gained attention. Several scholars are interested in refining these methods by including more criteria to achieve the most accurate and effective outcomes. Fractional derivative operators have been defined in a variety of ways over the years in the literature. The most well-known are: The Erdélyi-Kober fractional derivative [10], Riemann–Liouville fractional derivative [11], Caputo fractional derivative [12], and Caputo–Hadamard fractional derivative [13]. Using fractional calculus, several scholars have modeled a wide range of real-world processes, often with remarkable outcomes. Fredholm-Volterra type integral equations [14, 15] are often explored and solved using fractional differential equations because of their ability to capture memory and nonlocal effects [16]. In voice signal modeling, for instance, fractional calculus provides an alternative to traditional linear predictive coding by employing fractional-order integrals as basis functions [17]. By doing this, the number of parameters required is reduced, while the quality of the signal representation is improved; not only in signal processing, but in many other scientific domains as well [18]. PINNs [19], which are general feedforward neural networks connected to a redefined LF that takes into account initial/boundary conditions as well as the physical information from PDEs, are among the most representative NN-based PDE solvers. The standard PINN still faces several difficult issues despite its successful application in many scientific and engineering domains. [20] These include the high computational cost of solving PDEs using deep neural network structures, which are essentially a nonlinear function approximator; high wavenumber/convection problems; infinite domain problems; and limited computational efficiency or difficulty in computing for long-time evolution problems. Moreover, other generalizations have been proposed by researchers [21], such as the ψ-conformable fractional derivative, the ψ-Caputo fractional derivative, and the ψ-fractional operators. Fractional differential equations have been solved using a variety of optimization strategies, such as the altered differential evolution algorithm, in addition to analytical and neural network-based approaches [22].

Additionally, Katugampola proposed the Caputo–Katugampola fractional derivative (CKFD) [23], which is a generalization of the Riemann–Liouville fractional derivative and the Caputo–Hadamard fractional derivatives. The Caputo–Katugampola fractional derivative has attracted significant attention due to its ability to unify and extend several classical fractional operators within a single framework. In contrast to standard fractional derivatives, it introduces an additional parameter that enhances the flexibility in modeling nonlocal memory and scaling effects in complex systems. This makes it effective for describing phenomena where classical integer orders are insufficient. Such scenarios include anomalous diffusion, viscoelastic materials, biological processes, and complex dynamical systems, where memory-dependent behavior is essential. Consequently, the Caputo-Katugampola operator improves modeling accuracy and provides a more realistic representation of physical systems.

Machine learning has been transformed by neural networks, which make it possible to model intricate, non-linear interactions in a variety of areas [24]. These models, which draw inspiration from the composition and operations of biological neurons, have demonstrated impressive performance

in domains like image recognition, natural language processing, and reinforcement learning [25]. However, vanishing and ballooning gradients are problems that deeper networks encounter, making training more difficult as the network depth rises. The development of P-inf. machine learning introduced subject expertise as well as physical restrictions into models based on data [26]. This approach enhances generalization and model interpretability while reducing the need for vast volumes of data. PINNs have been effectively used to tackle data assimilation tasks, partial differential equations [27], and inverse problems [28, 29]. Such types of problems are combined with GEPINN, which incorporates grammatical evolution with PINN and solves some different types of equations like the Lane-Emden equation [30, 31]. The core elements of all deep neural networks [32, 33] are a linear transformation and a nonlinear A/F. To successfully train the deep network, the A/F plays a vital role, and the most widely used and efficient A/F is the Rectified Linear Unit (ReLU), denoted by the equation $f(x) = \max(x, 0)$. The introduction of ReLU made a significant advancement possible for fully supervised training of state-of-the-art deep networks. When the ReLU's input is positive, the gradient can spread, making ReLU-based networks easier to optimize than sigmoid or tanh units. Then, after further research on more activation functions, Swish or SiLu activation functions were found to work better than the ReLU A/F, which is defined as $f(x) = x * \text{sigmoid}(\beta x)$, where β is a constant or trainable parameter. In challenging domains, including image classification and machine translation, Swish continually exceeds ReLU when used on deep networks.

The NN method, especially PINN, has shown great potential for approximating the solution of FDEs. However, most of these methods are restricted to Caputo and Riemann–Liouville fractional derivatives. This limits their potential for solving generalized fractional dynamics.

Additionally, applying these NN approaches to the C–K fractional derivative is nontrivial because of the general form of its kernel and another scaling factor, making it more difficult to create models, automatically differentiate them, and ensure their numerical stability during training while enabling a more flexible and generalized approach toward modeling fractional behavior; however, NN-based approaches to the C–K fractional derivative have not been widely researched. Additionally, the Volterra integral equation integrated along with efficient quadrature methods in NN models has not been adequately explored.

To address these gaps, we introduce a unified framework for NN that is based on the C–K fractional derivative, which is implemented for data-driven (NFDE) and physics-informed (PINFDE) methods. The proposed framework uses a quadrature-based Volterra integral approach that is efficient in dealing with the nonlocal nature of the fractional derivative. Moreover, the proposed NN framework incorporates the SiLU A/F to solve the fractional system under consideration. This framework extends the capability of NNs for dealing with fractional dynamics in a more unified and efficient manner for solving complex FDEs.

The proposed framework extends the capability of NN to model fractional systems in a more generalized, accurate, and computationally efficient manner, thereby providing a robust tool for solving complex FDEs arising in scientific and engineering applications.

In this study, NFDE and PINFDE models are developed using the C–K fractional derivative and the SiLU activation function. The NFDE model is purely data-driven and relies only on observed data for training, while the PINFDE model integrates an additional P-inf. residual term derived from the governing equation. This enables the PINFDE model to enforce the underlying physical law during training, resulting in improved accuracy and stability as compared to the NFDE model.

The key contribution of the proposed work is outlined below:

- The problem is modeled through a formulation based on the C–K fractional derivative.
- The fractional model is solved using two computational approaches, NFDE and PINFDE.
- A Volterra integral formulation with quadrature discretization is employed to improve numerical accuracy.
- The SiLU A/F is used to improve the convergence and smoothness of the NN solution.
- A comparative analysis based on performance metrics (MSE, MAE, NSE, SSE, R^2) demonstrates the effectiveness of the proposed method.

The following is the structure of the paper: In Section 2, we present the preliminaries, including the definitions employed throughout the paper. In Section 3, we outline the main results, organized into three subsections: Subsection 3.1 entails NFDEs, Subsection 3.2 contains PINFDEs along with the neural network architecture, and Subsection 3.4 provides the flowchart of the proposed methodology. In Section 4, we describe the numerical methods, divided into three subsections: Subsection 4.1 provides quadrature-based Volterra integral equations to demonstrate the practical implementation of the proposed results, Subsection 4.2 presents the performance metrics used to assess the efficacy of the model, and Subsection 4.3 highlights the SiLU A/F, which yields improved accuracy. Section 5 contains the numerical experiments, where four examples are discussed; notably, Example 5.1 provides a comparison with the existing NFDE and PINFDE approximation. Finally, Section 6 provides the discussion and conclusion of the manuscript.

2. Preliminaries

In this section, we present the basic mathematical concepts and definitions required to develop the proposed method.

Definition 2.1. *[16] (Caputo fractional derivative). The Caputo derivative of order $\gamma > 0$ for the function $F(t)$, $t \in (c, d)$ is defined as*

$$
\begin{aligned}
{}_cD_t^{\gamma} F(t) &= {}_cI_t^{(m-\gamma)}\left[F^{(m)}(t)\right] \\
&= \frac{1}{\Gamma(m-\gamma)} \int_c^t (t-s)^{m-\gamma-1} F^{(m)}(s) ds,
\end{aligned}
$$

where m is a positive integer satisfying $m - 1 < \gamma \leq m$.

Definition 2.2. *[16] (Riemann–Liouville fractional integral). The Riemann–Liouville integral with order $\gamma > 0$ of the given function $F(t)$, $t \in (c, d)$ is defined as*

$$
{}_cI_t^{\gamma} F(t) = \frac{1}{\Gamma(\gamma)} \int_c^t (t-s)^{\gamma-1} F(s) ds,
$$

where $\Gamma(.)$ is the Euler's gamma function.

Definition 2.3. *[34] (Caputo–Katugampola fractional derivative) Let F be a function on $[0, \infty] \times \mathbb{R}$. The Katugampola type of fractional derivative with order $\gamma \in (0, 1)$ for a function F is defined by:*

$$
{}^{CK}D_t^{\gamma,\beta} F(t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \left(\frac{t^{\beta} - s^{\beta}}{\beta}\right)^{-\gamma} \frac{d}{ds} F(s) ds.
$$

Parameter $\beta \in (0, 1]$ is a tunable scaling parameter that modifies the time scale of the fractional operator. We can reduce it into the Caputo derivative when $\beta \to 1$,

$$\lim_{\beta \to 1} {}^{CK}D_t^{\gamma,\beta} F(t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \left(t - s\right)^{-\gamma} \frac{d}{ds} F(s) ds = {}_cD_t^{\gamma} F(t).$$

Definition 2.4. *[35] (Volterra-integral equation) Given the Volterra integral equation*

$$F(t) = f(t) + \int_0^t K(t, s)\, F(s)\, ds, \quad t \in [0, T]$$

with f continuous and kernel $K(t, s)$ continuous and bounded, its discretized form w.r.t. the equispaced partition

$$0 = t_0 < t_1 < \cdots < t_n = T,$$

is given for $l = 1, 2, 3, \cdots, n$ by

$$F(t_l) = f(t_l) + \sum_{j=0}^{l-1} \int_{t_j}^{t_{j+1}} K(t_l, s)\, F(s)\, ds.$$

3. Methodology

In the NFDE model, the NN is used to approximate the unknown non-linear function in a data-driven manner. On the other hand, the PINFDE framework uses the governing FDE to form the LF by adding a residual term, thereby satisfying the physical law. The main contribution and methodologies of the proposed work are explained in this section.

3.1. Neural fractional differential equation (NFDEs)

Deep learning methods and fractional calculus are used in NFDEs, where NNs are trained to approximate complicated or unknown components in systems controlled by fractional differential operators. These models work especially well for simulating memory and genetic influences that are present in a real-world system. A general scalar nonlinear FDE has the following expression:

$$ {}^{CK}D_t^{\gamma,\beta} x(t) = f(t, x(t)), t \in [t_0, T], \tag{3.1}$$

$$ x(t_0) = x_0, \tag{3.2}$$

where, ${}^{CK}D_t^{\gamma}$ is a Caputo–Katugampola fractional derivative operator; $x(t)$ is the state variable; $f(t, x(t))$ is a nonlinear function; γ is the fractional order; and x_0 is the initial condition.

Using a deep NN, the unknown (or complex) part of the dynamics $f(t, x)$ can be learned or approximated by representing $f(t, x) \approx \mathcal{N}_\theta(t, x)$, and the exact solution for $\gamma < 1$ is generally not available, so x_θ is considered an approximate solution. The approximate solution is obtained using a fractional numerical method that includes the NN form $\mathcal{N}_\theta(t, x)$ to describe the unknown dynamics. Hence, x_θ represents a hybrid approximation that includes the NN learning and physics-based fractional integration.

$$ {}^{CK}D_t^{\gamma,\beta} x_\theta(t) = \mathcal{N}_\theta(t, x_\theta(t)), \ t \in [t_0, T], \tag{3.3}$$

$$x_\theta(t_0) = x_0,$$

where θ is the set of NN parameters (weights and biases) learned to approximate the system dynamics.

The following is an expression for a general solution of the proposed NFDE using the C–K fractional derivative:

$$x_\theta(t) = x_0 + \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta} \right)^{\gamma-1} \mathscr{N}_\theta(s, x_\theta(s))\, ds. \tag{3.4}$$

The kernel $(t_{l+1}^\beta - s^\beta)^{\gamma-1}$ exhibits a weak singularity as $j \to l$. This is handled numerically by evaluating the corresponding term using stable discretization, ensuring that the contribution remains finite. In practice, the singular behavior is integrable and does not lead to numerical instability.

3.1.1. Derivation of proposed NFDE solution

The C–K fractional derivative is defined by

$${}^{CK}D_{0+}^{\gamma,\beta} x(t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta} \right)^{-\gamma} x'(s)\, ds.$$

The corresponding Katugampola fractional integral operator is defined by

$$I_{0+}^{\gamma,\beta} f(t) = \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta} \right)^{\gamma-1} f(s)\, ds. \tag{3.5}$$

The fundamental property of Caputo-type fractional operators for $(0 < \gamma < 1)$, is given as:

$$I_{0+}^{\gamma,\beta} \left[{}^{CK}D_{0+}^{\gamma,\beta} x(t) \right] = x(t) - x(0). \tag{3.6}$$

Applying the operator $(I_{0+}^{\gamma,\beta})$ to both sides of Eq (3.3), we obtain

$$I_{0+}^{\gamma,\beta} \left[{}^{CK}D_{0+}^{\gamma,\beta} x_\theta(t) \right] = I_{0+}^{\gamma,\beta} \left[\mathscr{N}_\theta(t, x_\theta(t)) \right].$$

Using the fundamental property, Eq (3.6), this reduces to

$$x_\theta(t) - x_0 = I_{0+}^{\gamma,\beta} \left[\mathscr{N}_\theta(t, x_\theta(t)) \right].$$

Substituting the definition of the fractional integral (3.5), we obtain

$$x_\theta(t) - x_0 = \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta} \right)^{\gamma-1} \mathscr{N}_\theta(s, x_\theta(s))\, ds.$$

Therefore, the solution becomes,

$$x_\theta(t) = x_0 + \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta} \right)^{\gamma-1} \mathscr{N}_\theta(s, x_\theta(s))\, ds.$$

Now, we define the LF to optimize the model:

$$L\Big(x_\theta(t)\Big) = L\Big(x_0 + \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta} \right)^{\gamma-1} \mathscr{N}_\theta(s, x_\theta))\, ds\Big).$$

The gradient of the LF, $dL/d\theta$, w.r.t. the parameters is then computed. The parameters θ are updated using the ADAM optimization algorithm, which is applied to minimize the LF L. By reducing the discrepancies between the expected and actual values, this iterative optimization procedure enhances the proposed NFDE's overall accuracy and effectiveness in capturing the dynamics of the system.

3.1.2. Neural network representation

Let $\mathcal{N}_\theta : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ with $\mathscr{L}$ layers and parameters $\theta = \{\mathscr{W}, \mathscr{B}\}_{l=1}^{\mathscr{L}}$, where

$\mathbb{R}$	Set of real numbers (e.g., time variable t).
$\mathbb{R}^d$	d-dimensional real vector space representing the state variable x.
$N_\theta : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$	Neural network parameterized by θ that takes (t, x) as input and outputs a d-dimensional vector representing the learned dynamics.
θ	Set of trainable parameters (weights $\mathscr{W}$ and biases $\mathscr{B}$) of the neural network.

For an input t, the NN output is represented as:

$$\begin{aligned} h^0 &= t \\ \mathscr{Z}^l &= \mathscr{W}^l h^{l-1} + \mathscr{B}^l, \;\; l = 1, 2, \cdots, \mathscr{L} - 1 \\ h^l &= \phi(\mathscr{Z}^l) \\ \mathcal{N}_\theta &= \mathscr{W}^{\mathscr{L}} h^{\mathscr{L}-1} + \mathscr{B}^{\mathscr{L}}, \end{aligned}$$

where h^l represents the output of the l-th hidden layers; $\mathscr{W}^l$ and $\mathscr{B}^l$ are trainable weight matrices and bias vectors; and $\phi(.)$ is a nonlinear activation function.

In this paper, we use the SiLU activation function, which is defined as

$$\phi(\mathscr{Z}) = \mathscr{Z}.\mathscr{S}(\mathscr{Z}).$$

Here:

- $\mathscr{Z}$ is the input,
- $\mathscr{S}(\mathscr{Z}) = \dfrac{1}{1 + e^{-\mathscr{Z}}}$ is the sigmoid function.

Thus, we expand the NN layers as:

$$\begin{aligned} h^1 &= \phi(\mathscr{W}^1 t + \mathscr{B}^1) \\ h^2 &= \phi(\mathscr{W}^2 h^1 + \mathscr{B}^2) \\ &\vdots \\ h^{\mathscr{L}-1} &= \phi(\mathscr{W}^{\mathscr{L}-1} h^{\mathscr{L}-2} + \mathscr{B}^{\mathscr{L}-1}) \\ \mathcal{N}_\theta &= \mathscr{W}^{\mathscr{L}} h^{\mathscr{L}-1} + \mathscr{B}^{\mathscr{L}}, \end{aligned}$$

hence, the nonlinear SiLU activation function enables $\mathcal{N}_\theta$ to represent the highly nonlinear activation function that can approximate the fractional derivative ${}^{CK}D_t^\gamma$.

3.2. PINFDEs

A detailed description of proposed PINFDEs is provided in this subsection. The approach combines NNs with ideas from physics to represent systems having fractional derivative characteristics. The description of dynamical systems is enhanced by incorporating the long-range dependencies and memory effects observed in fractional calculus with the representational power and adaptability of NNs because it ensures that the dynamics learned are in accordance with fundamental physical laws,

considering domain knowledge is crucial for these systems, leading to more precise and understandable models. Many real-world systems require the adaptation of fractional differential equations from conventional differential equations to represent intricate dynamics involving long-range dependence and memory effects. More precise modeling and prediction of system behaviors are made possible by this expansion, especially in situations when conventional differential equations might not be an adequate solution. To accommodate physics-based constraints, we integrate domain knowledge by incorporating new terms into the framework of NFDEs. Using the state vector in the NN results in the following modified equation:

$$\begin{aligned} {}^{CK}D_t^{\gamma,\beta} x_\theta(t) &= \mathscr{N}_\theta(t, x_\theta(t)) + f_{phy}(t, x_\theta(t)),\ t \in [t_0, T], \\ x_\theta(t_0) &= x_0, \end{aligned} \tag{3.7}$$

where f_{phy} is represented by the physics constraint where state vectors are impacted by the nonlinearity. The general solution of Eq (3.7) is given as:

$$x_\theta(t) = x_0 + \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta}\right)^{\gamma-1} \Big(\mathscr{N}_\theta(s, x_\theta(s)) + f_{phy}(s, x_\theta(s))\Big) ds.$$

The function of $f_{phy}(s, x_\theta(s))$ in Eq (3.7) is essential for understanding and representing the connections between the real dynamics and physics-based concerns. This illustrates the system's existing but constrained knowledge. Incorporating P-inf. variables into our neural fractional differential equations framework enables the NN to learn residual terms, which is one of the many benefits of our suggested methodology.

3.2.1. PINN representation

In the previous section, we discuss the representation of NNs; here, we explain how Physics-Informed neural networks (PINNs) work.

Description of symbols used in the PINN loss formulation.

Symbol	Meaning
N_f	Number of collocation (residual) points used to enforce the fractional differential equation.
N_d	Number of data points used to compute the supervised data loss term.
y_m	Exact or observed data value at location t_m.
$\|x(t_m) - y_m\|^2$	Squared difference between the network prediction and the observed data (data misfit error).
λ_{IC}	Weighting parameter controlling the relative contribution of the initial/boundary condition loss.
λ_{data}	Weighting parameter balancing the contribution of the data term with the physics-based residual.

The governing fractional equation can be expressed in integral form as:

$$x_\theta(t) = x_0 + \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta}\right)^{\gamma-1} \Big(\mathscr{N}_\theta(s, x_\theta(s)) + f_{phy}(s, x_\theta(s))\Big)\, ds.$$

Define the kernel function:

$$K(t, s) = \left(\frac{t^\beta - s^\beta}{\beta}\right)^{\gamma-1}.$$

Substituting the NN approximation into the governing equation, the continuous residual functional is defined as:

$$\mathscr{R}_\theta[x](t) = x_\theta(t) - x_0 - \frac{1}{\Gamma(\gamma)} \int_0^t K(t, s) \left(\mathscr{N}_\theta(s, x_\theta(s)) + f_{phy}(s, x_\theta(s))\right) ds.$$

Let the computational domain be discretized as:

$$0 = t_0 < t_1 < t_2 < \cdots < t_N = T.$$

The integral term is approximated using numerical quadrature:

$$\int_0^{t_n} K(t_n, s)\, g(s)\, ds \approx \sum_{j=0}^{n-1} \mathscr{W}_{n,j}\, g(t_j),$$

where the quadrature weights satisfy:

$$\mathscr{W}_{n,j} \approx K(t_n, t_j)\, \Delta t_j.$$

The discrete residual at each collocation point t_n is given by:

$$\mathscr{R}_n(\theta) = x_\theta(t_n) - x_0 - \frac{1}{\Gamma(\gamma)} \sum_{j=0}^{n-1} \mathscr{W}_{n,j} \left(\mathscr{N}_\theta(t_j, x_\theta(t_j)) + f_{phy}(t_j, x_\theta(t_j))\right).$$

Residual loss

$$L_{\text{res}}(\theta) = \frac{1}{N_f} \sum_{n \in \mathcal{F}} |\mathscr{R}_n(\theta)|^2.$$

Initial condition loss

$$L_{\text{IC}}(\theta) = |x_\theta(t_0) - x_0|^2.$$

Data loss

$$L_{\text{data}}(\theta) = \frac{1}{N_d} \sum_{m=1}^{N_d} |x_\theta(t_m) - y_m|^2.$$

3.2.2. Total loss function

The total loss function is defined as a weighted sum of all components:

$$L(\theta) = \underbrace{\frac{1}{N_f} \sum_{n \in \mathcal{F}} |\mathscr{R}_n(\theta)|^2}_{\text{Residual Loss}} + \underbrace{\lambda_{\text{IC}} |x_\theta(0) - x_0|^2}_{\text{Initial Condition Loss}} + \underbrace{\lambda_{\text{data}} \frac{1}{N_d} \sum_{m=1}^{N_d} |x_\theta(t_m) - y_m|^2}_{\text{Data Loss}}.$$

3.3. Optimization problem

The optimal network parameters are obtained by minimizing the total loss:

$$\theta^* = \arg\min_{\theta} L(\theta).$$

Network architecture

The NN design utilized in this study approximates the solution of fractional differential equations by integrating data and controlling physical rules into the learning process. It follows the standard architecture of a PINN. The components of the architecture are as follows, as shown in Figure 1:

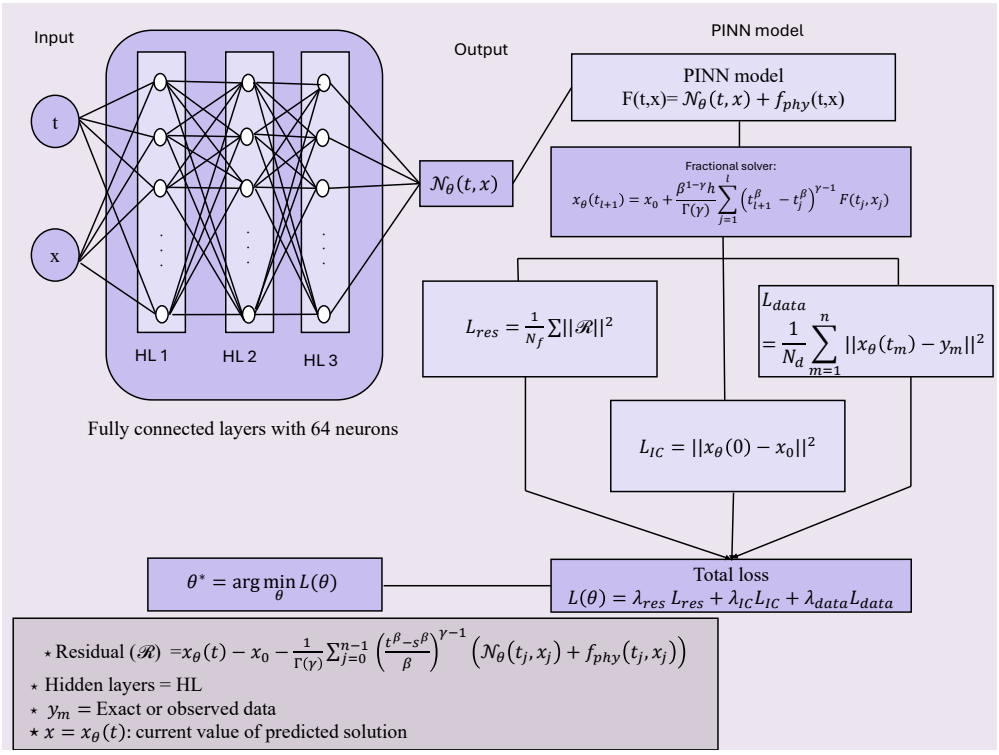


Figure 1. Neural network architecture for NFDE and PINFDE models.

• Input layer:

The input layer shows the spatial variable x and time domain t, which shows the input layer of governing fractional differential equations.

• Hidden layers:

A number of fully connected layers form the NN's backbone. Each HL consists of many neurons that use a nonlinear A/F to capture complex patterns and nonlinearities in the solution space.

• Ouput layer:

The output of the network is a single scalar function, which is the neural approximation of the fractional differential equation.

• Loss function:

Data loss:

This term enforces agreement with available initial and boundary conditions or observational data. It

is computed as the MSE between the predicted output and known values.

PINN loss:

This term ensures that the underlying differential equation is satisfied. It is obtained through the minimization of residuals and the substitution of the NN output $u(x, t)$ into the governing equation.

Total loss:

Combines both losses to train the network:

$$L_{total} = \lambda_1 L_{data} + \lambda_2 L_{PINN}.$$

3.4. Flowchart of the proposed work

In this subsection, we present the operational framework of the NN used to solve fractional differential equations, illustrated through a detailed flowchart in Figure 2.

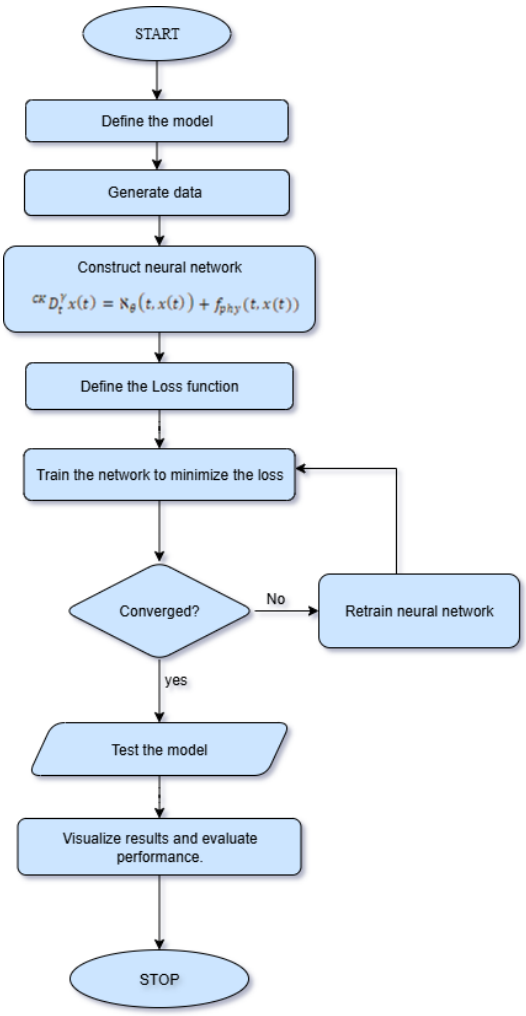


Figure 2. Flowchart demonstrating the model of fractional differential equation using neural networks.

The detailed explanation of the process is as follows:

(1) First, specify the nonlinear fractional differential equation, the problem domain, and the associated initial and boundary conditions. Additionally, give fractional parameters like γ and β values.
(2) Generate training data: Sample collocation, initial, and boundary points in the domain.
(3) Build the deep NN with the appropriate number of neurons per layer and HLs. Select the proper A/F. Before training starts, set the network's weights and biases at random.
(4) Define the LF for P-inf. and standard NN.
(5) Train the model using an optimization method, such as Adam. The training process lowers the overall loss by iteratively modifying the network weights. Continue training for an adequate number of epochs or until the model converges to a low-loss solution.
(6) Retrain the model by altering the architecture, hyperparameters, or training data distribution if it has not converged (i.e., the loss is still large or swings); if convergence is achieved, move on to the following phase. The RMSE, or total loss, can be used to measure convergence.
(7) After training is successful, test the model on unseen fractional orders (e.g., $\gamma = 0.95, 0.9, 0.8, \cdots$) to see how well it generalizes. Compare the model's predictions with the precise or benchmark solutions to evaluate its performance.
(8) Plot the exact solutions, predicted solutions, and absolute error surfaces accordingly. Calculate metrics of performance like MSE or RMSE. Using the C–K derivative, these studies and visualizations demonstrate the accuracy and reliability of the proposed NFDE and PINFDE model.

4. Numerical method

4.1. Quadrature-based Volterra integral equations

A numerical method based on the Volterra integral formulation is presented, which incorporates the C–K derivative into conventional fractional operators. This formulation facilitates a more flexible representation of memory effects and scaling aspects through the use of parameters γ and β, making it suitable for describing complex fractional dynamics. To solve the resulting fractional differential equations numerically, we use a convolution-type quadrature, which uses a sum over historical data points to discretize the fractional integral kernel. At each time step t_{l+1}, the update rule for the solution is provided by:

$$x(t_{l+1}) = x_0 + \frac{\beta^{1-\gamma} h}{\Gamma(\gamma)} \sum_{j=1}^{l} (t_{l+1}^{\beta} - t_j^{\beta})^{\gamma-1} F(t_j, x_j), \tag{4.1}$$

where, the data-driven NN approximation of the underlying nonlinear function governing the system is represented by $F(t_j, x_j)$, where h stands for the uniform time step.

This method is modified to account for the C–K operator's generalized time-scaling effects. A history-dependent weighting is naturally introduced by the kernel $(t_{l+1}^{\beta} - t_j^{\beta})^{\gamma-1}$, illustrating the solution's long-memory characteristics. The function $F(t_j, x_j)$ is not computed analytically in the context of NNs; instead, it is learned from data or approximated through training, either with PINNs or residual-based frameworks. Because neural approximators are used, the fractional numerical technique is better suited to depict systems in which the exact form of F is highly nonlinear.

4.1.1. Derivation of quadrature-based Volterra integral equations

In this subsection, the derivation of the solution presented in Eq (4.1) is provided. The Volterra integral formulation is first introduced, followed by quadrature-based discretization. Finally, the solution is obtained by integrating these approaches.

Volterra integral representation:

The C–K fractional differential equation in Eq (3.1) is given as:

$$^{CK}D^{\gamma,\beta}x(t) = f(t, x(t)).$$

Applying the corresponding fractional integral operator given in Eq (3.5), we have

$$x(t) = x_0 + \frac{1}{\Gamma(\gamma)} \int_0^t \left(\frac{t^\beta - s^\beta}{\beta}\right)^{\gamma-1} f(s, x(s))\, ds.$$

This is the type of nonlinear Volterra integral equation of the second kind, which can be written in the general form

$$x(t) = x_0 + \int_0^t K(t, s) f(s, x(s))\, ds,$$

where the kernel $K(t, s)$ is given by

$$K(t, s) = \left(\frac{t^\beta - s^\beta}{\beta}\right)^{\gamma-1}.$$

Quadrature-based discretization:

For the numerical approximation of the Volterra integral, the time domain is discretized into uniform nodes $t_j = jh$, where h represents the step size. Evaluating at t_{l+1}, the following expression is obtained:

$$x(t_{l+1}) = x_0 + \frac{1}{\Gamma(\gamma)} \int_0^{t_{l+1}} K(t_{l+1}, s) f(s, x(s))\, ds.$$

The integral is decomposed as

$$\int_0^{t_{l+1}} = \sum_{j=1}^{l} \int_{t_j}^{t_{j+1}}.$$

Thus,

$$x(t_{l+1}) = x_0 + \frac{1}{\Gamma(\gamma)} \sum_{j=1}^{l} \int_{t_j}^{t_{j+1}} K(t_{l+1}, s) f(s, x(s))\, ds.$$

Assuming $f(s, x(s))$ varies slowly over each subinterval, it is approximated as

$$f(s, x(s) \approx f(t_j, x_j)), \quad s \in [t_j, t_{j+1}],$$

which yields:

$$x(t_{l+1}) \approx x_0 + \frac{1}{\Gamma(\gamma)} \sum_{j=1}^{l} f(t_j, x_j) \int_{t_j}^{t_{j+1}} K(t_{l+1}, s) ds, \tag{4.2}$$

where

$$K(t,s) = \left(\frac{t^\beta - s^\beta}{\beta}\right)^{\gamma-1} = \beta^{1-\gamma}(t^\beta - s^\beta)^{\gamma-1},$$

To evaluate the following integral:

$$\int_{t_j}^{t_{j+1}} (t_{l+1}^\beta - s^\beta)^{\gamma-1} ds.$$

Let

$$u = t_{l+1}^\beta - s^\beta$$

then

$$ds = -\frac{1}{\beta s^{\beta-1}} du.$$

Approximating ($s \approx t_j$) withing the subinterval, we get

$$\int_{t_j}^{t_{j+1}} (t_{l+1}^\beta - s^\beta)^{\gamma-1} ds \approx \frac{1}{\beta t_j^{\beta-1}} \int_{u_{j+1}}^{u_j} u^{\gamma-1} du.$$

Evaluating the integral,

$$\int u^{\gamma-1} du = \frac{u^\gamma}{\gamma},$$

we get

$$\approx \frac{1}{\beta t_j^{\beta-1}(\gamma)} \left[(t_{l+1}^\beta - t_j^\beta)^\gamma - (t_{l+1}^\beta - t_{j+1}^\beta)^\gamma\right]. \tag{4.3}$$

The above expression (4.3) gives:

$$\frac{1}{\beta t_j^{\beta-1}(\gamma)} \left[(t_{l+1}^\beta - t_j^\beta)^\gamma - (t_{l+1}^\beta - t_{j+1}^\beta)^\gamma\right].$$

Let

$$A = t_{l+1}^\beta - t_j^\beta, \quad B = t_{l+1}^\beta - t_{j+1}^\beta,$$

then

$$A - B = t_{j+1}^\beta - t_j^\beta.$$

Using a first order Taylor expansion of t^β about t_j, we have,

$$t_{j+1}^\beta - t_j^\beta \approx \beta t_j^{\beta-1}(t_{j+1} - t_j). \tag{4.4}$$

For a uniform time discretization,

$$h = t_{j+1} - t_j, \tag{4.5}$$

using Eqs (4.4) and (4.5), we get

$$t_{j+1}^\beta - t_j^\beta \approx \beta t_j^{\beta-1} h.$$

Substituting this into the kernel difference and using a first-order expansion,

$$A^\gamma - B^\gamma \approx (\gamma) A^{\gamma-1}(A - B),$$

we obtain

$$\int_{t_j}^{t_{j+1}} (t_{l+1}^{\beta} - s^{\beta})^{\gamma-1} ds \approx h(t_{l+1}^{\beta} - t_j^{\beta})^{\gamma-1}. \tag{4.6}$$

The kernel $(t_{l+1}^{\beta} - s^{\beta})^{\gamma-1}$ exhibits a weak singularity at $j \to l$. This is handled numerically by evaluating the corresponding term using stable discretization, ensuring that the contribution remains finite. In practice, the singular behavior is integrable and does not lead to numerical instability.

Substituting Eq (4.6) into Eq (4.2), we obtain

$$x(t_{l+1}) = x_0 + \frac{\beta^{1-\gamma} h}{\Gamma(\gamma)} \sum_{j=1}^{l} (t_{l+1}^{\beta} - t_j^{\beta})^{\gamma-1} F(t_j, x_j).$$

The final quadrature-based scheme becomes

$$\boxed{x(t_{l+1}) = x_0 + \frac{\beta^{1-\gamma} h}{\Gamma(\gamma)} \sum_{j=1}^{l} (t_{l+1}^{\beta} - t_j^{\beta})^{\gamma-1} F(t_j, x_j).}$$

4.1.2. Convergence and stability analysis

Proposition 4.1. *Let $F \in C^1([0,T] \times \mathbb{R})$ and let $x(t)$ be the exact solution. The local truncation error of the convolution quadrature scheme* (4.1) *satisfies:*

$$|\tau_l| \leq C_1 h^{\gamma},$$

where $C_1 > 0$ is independent of h.

Proof. The exact solution satisfies:

$$x(t_{l+1}) = x_0 + \frac{\beta^{1-\gamma}}{\Gamma(\gamma)} \int_0^{t_{l+1}} (t_{l+1}^{\beta} - s^{\beta})^{\gamma-1} F(s, x(s)) ds.$$

Defining local truncation error τ and replacing x_j by exact values $x(t_j)$, we get

$$\tau_{l+1} = \frac{\beta^{1-\gamma}}{\Gamma(\gamma)} \Big[\int_0^{t_{l+1}} K(t_{l+1}, s) F(s, x(s)) ds - h \sum_{j=1}^{l} K(t_{l+1}, t_j) F(t_j, x(t_j)) \Big],$$

where

$$K(t, s) = (t^{\beta} - s^{\beta})^{\gamma-1}.$$

Splitting the interval

$$[0, t_{l+1}] = \bigcup_{j=0}^{l} [t_j, t_{j+1}],$$

we get

$$\tau_{l+1} = C \sum_{j=0}^{l} \Big[\int_{t_j}^{t_{j+1}} K(t_{l+1}, s) F(s, x(s)) ds - hK(t_{l+1}, t_j) F(t_j, x(t_j)) \Big].$$

Separating the last interval, we obtain

$$\tau_{l+1} = C(\sum_{j=0}^{l-1} E_j + E_l),$$

where E_j represents the local error on $[t_j, t_{j+1}]$,

$$E_j = \int_{t_j}^{t_{j+1}} K(t_{l+1}, s)F(s, x(s))ds - hK(t_{l+1}, t_j)F(t_j, x(t_j)).$$

For $j < 1$, the kernel is smooth and bounded. Thus, quadrature estimates (using Taylor expansion) give:

$$|E_j| \leq Ch^2.$$

Hence,

$$\sum_{j=0}^{l-1} |E_j| \leq Ch.$$

Now, let

$$E_l = \int_{t_l}^{t_{l+1}} (t_{l+1}^{\beta} - s^{\beta})^{\gamma-1} F(s, x(s))ds.$$

Since F is bounded we have,

$$|F(s, x(s))| \leq M.$$

Thus,

$$|E_l| \leq C \int_{t_l}^{t_{l+1}} (t_{l+1}^{\beta} - s^{\beta})^{\gamma-1} ds.$$

For $s \in [t_l, t_{l+1}]$, using the mean value theorem, we get

$$(t_{l+1}^{\beta} - s^{\beta}) = \beta\xi^{\beta-1}(t_{l+1} - s), \quad \xi \in (s, t_{l+1}).$$

Hence,

$$(t_{l+1}^{\beta} - s^{\beta}) \sim C(t_{l+1} - s).$$

Therefore,

$$|E_l| \leq C \int_{t_l}^{t_{l+1}} (t_{l+1} - s)^{\gamma-1} ds.$$

Let us consider $u = (t_{l+1} - s)$, and we get

$$|E_l| = Ch^{\gamma}.$$

Now, we have

$$\sum_{j=0}^{l-1} |E_j| = O(h), \quad |E_l| = O(h^{\gamma}).$$

Since $0 < \gamma < 1$, the dominant term is h^{γ}. Thus,

$$\tau_{l+1} \leq Ch^{\gamma}.$$

□

Theorem 4.2. *(Convergence) Under the same assumptions as Proposition 4.1, with F Lipschitz in x with constant $L > 0$, the global error satisfies:*

$$\max_{1\le l\le N} |x(t_l) - x_l| \le Ch^{\gamma}$$

where C is independent of the step size h. Thus, the method converges with order $O(h^{\gamma})$ as $h \to 0$.

Proof. Let $e_l = x(t_l) - x_l$ denote the global error at step l. Subtracting the numerical scheme from the exact integral equation gives:

$$e_{l+1} = \frac{\beta^{1-\gamma}}{\Gamma(\gamma)}\Big[\int_0^{t_{l+1}} K(t_{l+1}, s)F(s, x(s))ds - h\sum_{j=1}^{l} K(t_{l+1}, t_j)F(t_j, x_j)\Big].$$

Then, adding and subtracting the term gives

$$h\sum_{j=1}^{l} K(t_{l+1}, t_j)F(t_j, x(t_j)).$$

Then we get

$$e_{l+1} = C\sum_{j=1}^{l} hK(t_{l+1}, t_j)\left[F(t_j, x(t_j)) - F(t_j, x_j)\right] + \tau_{l+1},$$

where τ_{l+1} is the local truncation error.

Taking absolute values and applying the Lipschitz condition:

$$|F(t_j, x(t_j)) - F(t_j, x_j)| \le L|e_j|,$$

we get,

$$|e_{l+1}| \le Ch\sum_{j=1}^{l} K(t_{l+1}, t_j)\,|e_j| + |\tau_{l+1}|.$$

From Proposition 4.1,

$$|\tau_{l+1}| \le Ch^{\gamma}$$

$$|e_{l+1}| \le Ch\sum_{j=1}^{l}(t_{l+1}^{\beta} - t_j^{\beta})^{\gamma-1}|e_j| + Ch^{\gamma}.$$

Apply the fractional Grönwall inequality. If

$$e_{l+1} \le Ch\sum_{j=1}^{l} w_{(l+1,j)}e_j + Ch^{\gamma},$$

then,

$$|e_l| \le Ch^{\gamma}.$$

Thus,

$$\max_{1\le l\le N} |x(t_l) - x_l| \le Ch^{\gamma}.$$

Hence, the method converges with order $O(h^{\gamma})$ as $h \to 0$. □

Theorem 4.3. *(Stability) Let $\mathcal{N}_\theta$ denote the NN approximation of F with uniform error*

$$\epsilon_{NN} = \sup_{t,x_\theta} |\mathcal{N}_\theta(t, x_\theta) - F(t, x_\theta)|.$$

Let x_l and $\tilde{x}_l$ denote the numerical solutions obtained using F and $\mathcal{N}_\theta$, respectively. Then, the error satisfies:

$$\max_{1\le l\le N} |\tilde{x}_l - x_l| \le C \cdot \epsilon_{NN},$$

where C is a constant independent of step size h.

Proof. Let $\delta_l = |\tilde{x}_l - x_l|$. From the numerical scheme

$$x_{l+1} = x_0 + Ch \sum_{j=1}^{l} K(t_{l+1}, t_j) F(t_j, x_j),$$

$$\tilde{x}_{l+1} = x_0 + Ch \sum_{j=1}^{l} K(t_{l+1}, t_j) \mathcal{N}_\theta(t_j, \tilde{x}_j).$$

We obtain

$$\delta_{l+1} \le Ch \sum_{j=1}^{l} K(t_{l+1}, t_j) \left| \mathcal{N}_\theta(t_j, \tilde{x}_j) - F(t_j, x_j) \right|.$$

Adding and subtracting with $F(t_j, \tilde{x}_j)$, we get

$$= Ch \sum_{j=1}^{l} K(t_{l+1}, t_j) \left(\left| \mathcal{N}_\theta(t_j, \tilde{x}_j) - F(t_j, \tilde{x}_j) \right| + \left| F(t_j, \tilde{x}_j) - F(t_j, x_j) \right| \right).$$

Using the Lipschitz continuity of F,

$$|F(t_j, \tilde{x}_j) - F(t_j, x_j)| \le L\delta_j,$$

and the definition of ϵ_{NN},

$$\left| \mathcal{N}_\theta(t_j, \tilde{x}_j) - F(t_j, \tilde{x}_j) \right| \le \epsilon_{NN},$$

we obtain

$$\delta_{l+1} \le Ch \sum_{j=1}^{l} K(t_{l+1}, t_j) \left(L\delta_j + \epsilon_{NN} \right).$$

Separating the two terms and applying the same weight bound as in Theorem 4.2, followed by the discrete Gronwall inequality, gives

$$\max_l \delta_l \le C \cdot \epsilon_{NN},$$

where C is independent of h. Hence, the scheme is stable with respect to the NN approximation error. □

4.2. Performance evaluation metrics

The precision and effectiveness of the proposed NN models for solving fractional differential equations are assessed using several standard performance metrics, such as MSE, MAE, SSE, R^2, and NSE.

The MSE is computed as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(u_{\text{NN}}^{(i)} - u_{\text{true}}^{(i)}\right)^2,$$

where u_{NN} and u_{true} denote the predicted and exact solutions, respectively, and N is the number of test points. The MAE measures the average magnitude of the absolute error, given by

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|u_{\text{NN}}^{(i)} - u_{\text{true}}^{(i)}\right|.$$

The SSE represents the total squared deviation from the ground truth:

$$\text{SSE} = \sum_{i=1}^{N}\left(u_{\text{NN}}^{(i)} - u_{\text{true}}^{(i)}\right)^2.$$

To evaluate the model's explanatory power, we use the coefficient of determination, defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(u_{\text{NN}}^{(i)} - u_{\text{true}}^{(i)})^2}{\sum_{i=1}^{N}(u_{\text{true}}^{(i)} - \bar{u}_{\text{true}})^2},$$

where $\bar{u}_{\text{true}}$ is the mean of the true solution. Last, NSE is calculated to quantify the predictive skill of the model:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{N}(u_{\text{NN}}^{(i)} - u_{\text{true}}^{(i)})^2}{\sum_{i=1}^{N}(u_{\text{true}}^{(i)} - \bar{u}_{\text{true}})^2}.$$

High R^2 and NSE values, as well as low MSE, MAE, and SSE values, indicate great model performance. All these measures show how well the suggested NFDE and PINFDE frameworks can approximate solutions to complex fractional-order systems.

4.3. SiLU activation function

The Sigmoid Linear Unit, more commonly known as SiLU, is a popular A/F used in NNs for its efficiency and performance. As a self-gated function, SiLU elegantly combines the sigmoid and rectified linear unit functions. It was introduced in the paper Searching for A/F, originally called Swish. It has unique properties, like smoothness and non-monotonicity, which enable it to outperform traditional A/F like ReLU for deep models, providing better accuracy and faster convergence during model training. SiLU is defined by multiplying an input value by its sigmoid. This self-gating mechanism enables the function to smoothly transition from being linear for positive inputs to near-zero for large negative inputs, which helps regulate the flow of information through the network. One key feature of SiLU is its non-monotonicity; it dips slightly below zero for small negative inputs before rising back toward zero. This property is believed to enhance NN's expressive power by creating a

richer gradient landscape and preventing the vanishing gradient problem, which can considerably slow or stop the learning process in deep architectures. Additionally, SiLU is smooth everywhere: This is an important positive feature since it guarantees smoothness in the gradient and enables optimization algorithms like gradient descent to work without problems. SiLU is defined as:

$$\text{SiLU(x)} = x.\mathscr{S}(x).$$

Here:
- x is the input,
- $\mathscr{S}(x) = \dfrac{1}{1+e^{-x}}$ is the sigmoid function.

We discuss the computational framework of our model in Algorithm 1.

Algorithm 1: NFDE–PINFDE Fractional Solution Framework

Input: Fractional orders $\{\gamma_k\}$; time domain $[0, T]$; step size h; initial condition x_0; neural networks $\mathcal{N}_\theta^{NF}$, $\mathcal{N}_\theta^{PF}$; epochs E; learning rate η

Output: Approximate solutions $x^{NF}(t)$, $x^{PF}(t)$ and trained parameters θ

Initialize $t_l = lh,\ l = 0, \ldots, N$, $x(0) = x_0$;

Normalize data and split into training/testing sets;

for *epoch* = 1 *to* E **do**

  for *each* γ_k **do**

    Compute fractional state:

$$x(t_{n+1}) = x_0 + \frac{h}{\Gamma(\gamma_k)} \sum_{j=1}^{n} \left(\frac{t_{n+1}^{\rho} - t_j^{\rho}}{\rho} \right)^{\gamma_k - 1} F_j$$

    if *NFDE model* **then**

      $F_j = \mathcal{N}_\theta^{NF}(t_j, x_j)$;

    else

      $F_j = \mathcal{N}_\theta^{PF}(t_j, x_j) + f_{phy}(t_j, x_j)$;

    Compute residual loss L;

  Average loss over all γ_k;

  Update parameters:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$$

for *each test* γ **do**

  Compute $x^{NF}(t)$ using trained NFDE model;

  Compute $x^{PF}(t)$ using trained PINFDE model;

Compute reference solution using numerical scheme;

Evaluate performance metrics (MSE, MAE, R^2, NSE);

return $x^{NF}(t)$, $x^{PF}(t)$, θ

5. Numerical results

From a computational perspective, the proposed NN-based approach involves an initial training cost, which depends on the network architecture and number of training epochs. However, once the model is trained, it can provide rapid evaluations of the solution at arbitrary points without requiring repeated numerical integration. In contrast, traditional numerical methods require recomputation for each new evaluation, which can be computationally expensive for repeated queries. Therefore, the proposed method offers an advantage in applications where fast and repeated solution evaluation is required. In our proposed method, the network is trained using 100 collocation points for 1000 epochs, which determine the overall training complexity. The improved performance of the PINFDE model can be attributed to the incorporation of physics-based constraints in the LF. These constraints guide the learning process and restrict the solution space to physically consistent solutions. This effect becomes more significant for fractional orders $\gamma < 1$, where the system exhibits memory-dependent behavior and increased complexity. In such cases, a purely data-driven approach (NFDE) may struggle to capture the underlying dynamics accurately. By incorporating the governing fractional equation through the residual term, the PINFDE model is better able to capture these effects, resulting in improved stability and accuracy compared to the NFDE model.

Example 5.1. *Consider the fractional differential equation of order* $0 < \gamma \leq 1$:

$$ {}^{CK}D^{\gamma,\beta}x(t) = -t - x - t^2x^2 + \frac{(te^{-t} + t^2e^{-3t})}{x}, $$

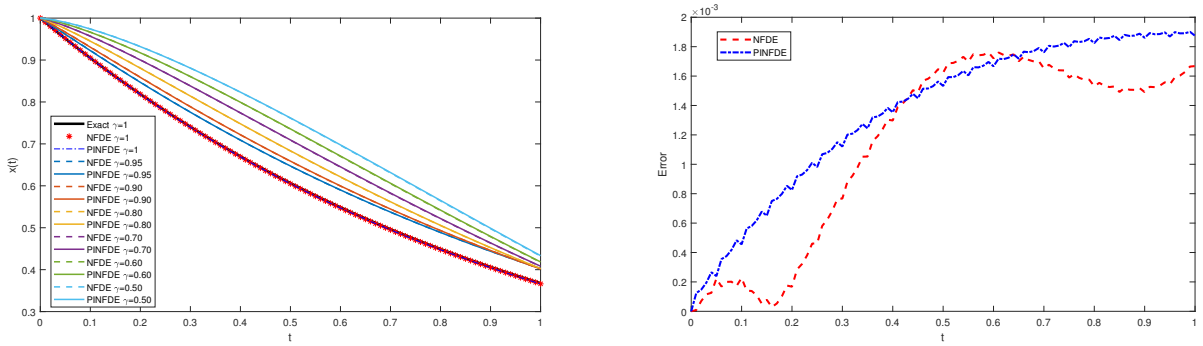
where $x_0 = 1$, $0 \leq t \leq 1$.

In this study, the fractional differential equation is solved using two approaches: A standard proposed NN framework (NFDE) and a proposed P-inf. NN framework (PINFDE). Both models are designed to approximate the solution of the fractional system within the period $[0, 1]$*, which is divided into one hundred uniform time periods. To capture the dynamics of the exact solution, these discrete points act as the training data. To strike a balance between gradient stability and computational performance, a mini-batch size of* 10 *is used during the training phase. The network design consists of three HLs with* 64 *neurons. To add nonlinearity and improve learning capacity, the Sigmoid Linear Unit (SiLU) A/F is used. With a fixed learning rate of 0.05, the Adam optimizer is used to train the system for one thousand epochs. For the baseline case, the network is trained at the integer-order setting* $\gamma = 1$*, and its generalization ability is subsequently tested for fractional orders* $\gamma = 0.95, 0.9, 0.8, \cdots$ *. In this study,* β *is taken as* 0.5 *to demonstrate the capability of the proposed framework. The proposed PINFDE formulation explicitly incorporates the governing physical law into the approximation by including the residual P-inf. term* $-x$*, with other terms approximated by the NN. This warrants that the learned solution will fit the data while satisfying the underlying physical constraints, thus improving generalization and stability. The proposed NFDE and PINFDE models in terms of accuracy, as shown in Table 1, demonstrate that for integer order at* $\gamma = 1$*, both NFDE and PINFDE solutions closely match the exact solution, confirming the accuracy of the proposed method.*

Table 1. Comparison table of proposed NFDE and PINFDE with the exact solution.

t	Exact [16]	Proposed NFDE	Proposed PINFDE
0	1	1	1
0.10	0.9048374180	0.9048335552	0.9043827652
0.20	0.8187307531	0.8184291124	0.8179075717
0.30	0.7408182207	0.7402055263	0.7397003173
0.40	0.6703200460	0.6694988608	0.6689702868
0.50	0.6065306597	0.6056125164	0.6050027012
0.60	0.5488116361	0.5478584766	0.5471508502
0.70	0.4965853038	0.4955883622	0.4948300123
0.80	0.4493289641	0.4482131600	0.4475114345
0.90	0.4065696597	0.4052139520	0.4047169685
1.00	0.3678794412	0.3661450147	0.3660140037

Figure 3a illustrates the predicted solutions of the proposed NFDE and PINFDE for different fractional orders $\gamma \in \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5\}$. *It can be observed that as the fractional order decreases, the solution profiles gradually change, reflecting the influence of memory effects inherent in fractional systems. A more specific comparison of error dynamics can be found in Figure 3b for* $\gamma = 1$, *which shows that the error values remain low throughout the domain, indicating high accuracy of the proposed models. Moreover, the PINFDE model exhibits comparatively lower error than the NFDE model, highlighting the advantage of incorporating P-inf. constraints in improving solution accuracy and stability. It is important to note that the error plots consistently show a smoother and more stable trajectory for the PINFDE model compared to the NFDE model. This behavior can be explained by the incorporation of P-inf. constraints in the LF. Specifically, the additional physics-based loss term acts as a regularization mechanism, guiding the NN to satisfy the underlying governing equations during training. As a result, the PINFDE model avoids overfitting to data and produces more physically consistent solutions, leading to reduced oscillations and smoother error profiles. In contrast, the NFDE model, being purely data-driven, may exhibit relatively larger fluctuations in error due to the absence of such constraints. This demonstrates that the inclusion of physical knowledge not only improves accuracy but also enhances the stability and generalization capability of the model.*



(a) Predicted solutions of the model at different values of γ

(b) Error analysis at $\gamma = 1$

Figure 3. Predicted solutions and error analysis for the NFDE and PINFDE models.

NFDE and PINFDE models closely follow the expected solution behavior, demonstrating their capability to effectively approximate the system dynamics across varying fractional orders, and as $\gamma < 1$, the exact solution is generally not available, so we make the comparison between our model and the numerical solution (L1) given in Figure 4. It can be seen from the figure that both models closely follow the numerical solution across all considered fractional orders, demonstrating the effectiveness in capturing the non-local dynamics of the system. In addition, the model of PINFDE demonstrates consistency with the numerical solutions in contrast to NFDE with more accuracy because of the inclusion of the P-inf. conditions.

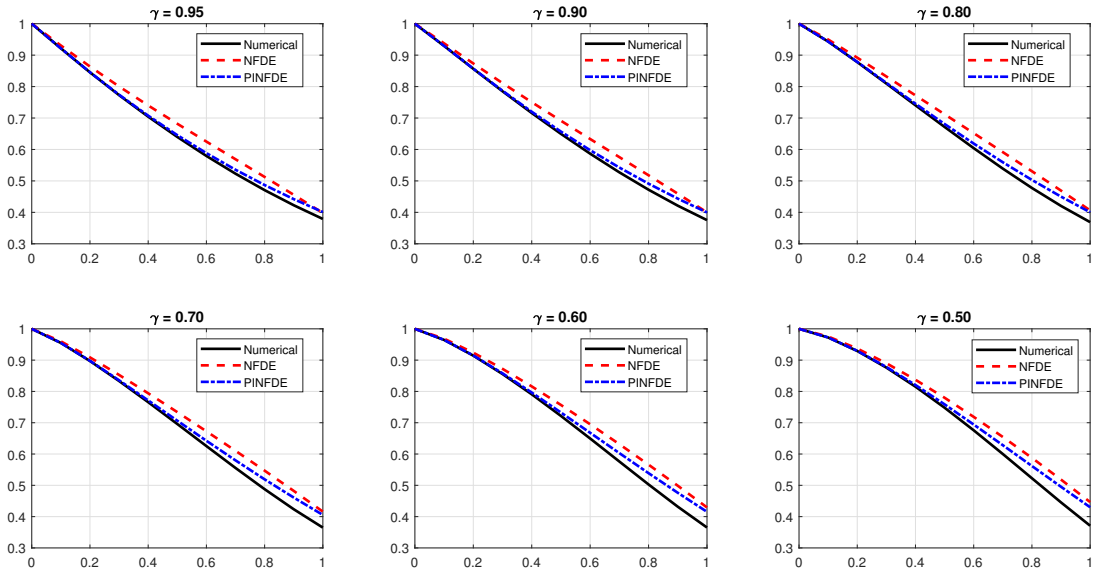


Figure 4. Comparison graph of the proposed NFDE and proposed PINFDE with the numerical solution at different values of $\gamma < 1$.

Figure 5 shows the error analysis for different values of $\gamma < 1$, where the error remains relatively small across the domain, confirming the stability and reliability of the proposed frameworks. In addition, the PINFDE model consistently produces lower error compared to the NFDE model, highlighting its better performance in approximating fractional-order systems. Figure 6 shows the convergence of total, data, and residual loss with respect to training epochs, where we can observe that all loss components decrease smoothly and consistently, indicating stable training of the proposed PINFDE model. The residual loss decreases steadily, confirming that the model effectively learns the governing fractional equation. Moreover, the smooth and non-oscillatory behavior of the loss curves highlights the regularization effect of the P-inf. loss term while also demonstrating the effectiveness of the SiLU A/F in improving convergence and training stability.

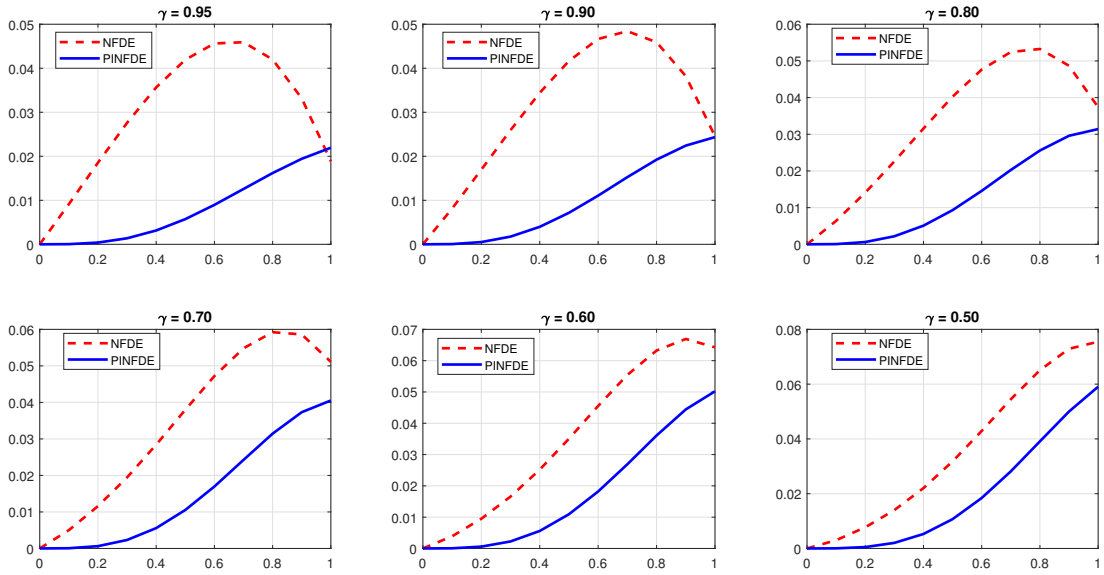


Figure 5. Error analysis at $\gamma < 1$.

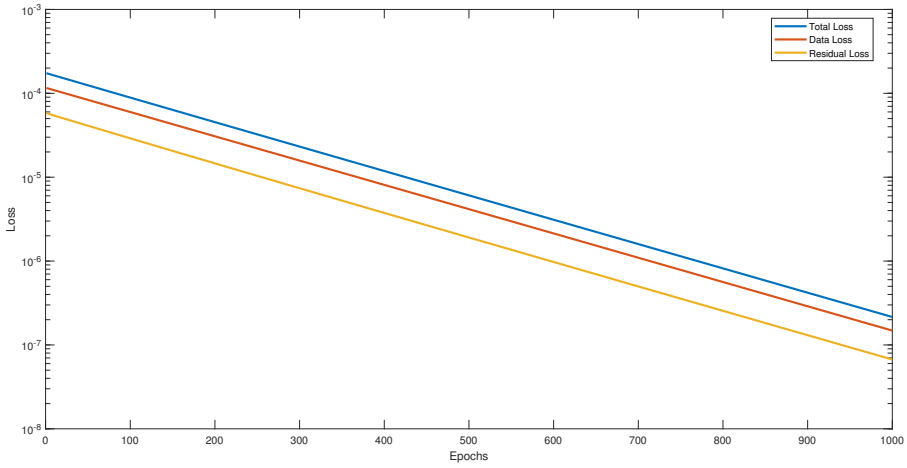


Figure 6. Convergence of Total, Data, and Residual loss during training.

The error analysis is presented in Tables 2 and 3 represents the results of comparative analysis for $\gamma < 1$. *The results confirm that the presented model preserves its accuracy when the fractional order is reduced, thus enabling it to capture the nonlocality of the problem. Performance analysis for the presented NFDE and the proposed PINFDE models is provided in the table, where the reduction of errors indicates better stability and accuracy for the models, in particular for PINFDE. The proposed PINFDE method outperforms the proposed NFDE method in every case, with much lower orders of error given in Table 4, highlighting its robustness and improved convergence characteristics. This illustrates the benefit of embedding physical knowledge within the learning framework, leading to lower approximation errors and more reliable predictions.*

Table 2. Performance analysis of the models at $\gamma = 1$.

Method	R^2	NSE	MAE	MSE	SSE
Proposed NFDE	0.99997	0.99997	7.3215×10^{-4}	7.78×10^{-7}	2.34×10^{-5}
Proposed PINFDE	1	1	1.3745×10^{-5}	4.3885×10^{-10}	1.316×10^{-8}

Table 3. Comparison table of numerical, NFDE, and PINFDE at $\gamma < 1$.

	$\gamma = 0.95$			$\gamma = 0.90$			$\gamma = 0.80$			$\gamma = 0.70$			$\gamma = 0.60$			$\gamma = 0.50$		
t	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE
0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	0.9215	0.9307	0.9216	0.9294	0.9375	0.9294	0.9433	0.9497	0.9433	0.9550	0.9600	0.9550	0.9646	0.9685	0.9646	0.9724	0.9754	0.9724
0.2	0.8452	0.8638	0.8456	0.8565	0.8735	0.8570	0.8778	0.8919	0.8785	0.8972	0.9088	0.8978	0.9143	0.9238	0.9149	0.9292	0.9370	0.9297
0.3	0.7728	0.8004	0.7741	0.7851	0.8110	0.7868	0.8096	0.8322	0.8118	0.8333	0.8528	0.8356	0.8556	0.8722	0.8578	0.8761	0.8901	0.8781
0.4	0.7043	0.7400	0.7075	0.7160	0.7504	0.7199	0.7404	0.7720	0.7455	0.7657	0.7941	0.7713	0.7909	0.8160	0.7965	0.8152	0.8372	0.8205
0.5	0.6399	0.6818	0.6456	0.6497	0.6913	0.6568	0.6716	0.7119	0.6810	0.6960	0.7339	0.7065	0.7217	0.7568	0.7327	0.7480	0.7798	0.7587
0.6	0.5793	0.6250	0.5883	0.5866	0.6332	0.5976	0.6041	0.6518	0.6187	0.6254	0.6726	0.6423	0.6495	0.6951	0.6677	0.6758	0.7187	0.6942
0.7	0.5229	0.5688	0.5355	0.5272	0.5756	0.5424	0.5391	0.5915	0.5593	0.5554	0.6102	0.5797	0.5760	0.6314	0.6029	0.6001	0.6545	0.6282
0.8	0.4706	0.5126	0.4868	0.4719	0.5178	0.4911	0.4775	0.5307	0.5030	0.4876	0.5468	0.5190	0.5027	0.5659	0.5388	0.5226	0.5877	0.5617
0.9	0.4226	0.4558	0.4420	0.4212	0.4594	0.4436	0.4204	0.4691	0.4500	0.4235	0.4822	0.4609	0.4317	0.4986	0.4761	0.4453	0.5182	0.4953
1.0	0.3790	0.3979	0.4010	0.3753	0.3999	0.3997	0.3688	0.4062	0.4003	0.3649	0.4160	0.4054	0.3652	0.4294	0.4154	0.3709	0.4464	0.4300

Table 4. Performance analysis of the models at $\gamma < 1$.

	NFDE					PINFDE				
γ	RMSE	MAE	SSE	R^2	NSE	RMSE	MAE	SSE	R^2	NSE
0.95	1.1311e-03	3.0681e-02	5.7688e-02	0.9666	0.9666	1.1811e-04	7.9270e-03	6.0237e-03	0.9965	0.9965
0.90	1.2219e-03	3.1663e-02	6.2316e-02	0.9652	0.9652	1.6180e-04	9.4091e-03	8.2518e-03	0.9954	0.9954
0.80	1.4238e-03	3.3432e-02	7.2616e-02	0.9615	0.9615	2.8049e-04	1.2365e-02	1.4305e-02	0.9924	0.9924
0.70	1.6302e-03	3.4728e-02	8.3139e-02	0.9574	0.9574	4.2955e-04	1.5031e-02	2.1907e-02	0.9888	0.9888
0.60	1.8057e-03	3.5359e-02	9.2089e-02	0.9532	0.9532	5.8514e-04	1.7141e-02	2.9842e-02	0.9848	0.9848
0.50	1.9169e-03	3.5241e-02	9.7763e-02	0.9495	0.9495	7.1872e-04	1.8540e-02	3.6655e-02	0.9811	0.9811

Example 5.2. *Consider the following fractional differential equation:*

$$ {}^{CK}D^{2\gamma}x(t) + \frac{2}{t}{}^{CK}D^{\gamma}x(t) + 2x(t) = 0, $$

where $x_0 = 1$, ${}^{CK}D^{\gamma,\beta}x_0 = 0$.

In this example, the problem is computed using proposed NFDE and proposed PINFDE approaches for fractional orders in the range $0 < \gamma \leq 1$. *In the proposed PINFDE model, variable x, representing a known physical term, is implemented as a structured and well-defined element of the system, while the remaining fractional-order terms are learned by the proposed NFDE part of the model, thereby combining data-driven learning with physics-based constraints. Performance is tested for different fractional orders, namely* $\gamma = 1, 0.95, 0.9,$ *and* 0.8. *A detailed comparison at* $\gamma = 1$ *is shown in Table 5, where NFDE and PINFDE models show results that closely match the expected solution, indicating high accuracy of the proposed model in the integer-order case, and performance metrics such as MSE, MAE, SSE,* R^2, *and NSE are shown in Table 6, where low error values confirm the stability and effectiveness of the models, with the PINFDE demonstrating improved performance due to the inclusion of P-inf. constraints. These results demonstrate that proposed PINFDE consistently improves solution accuracy compared to proposed NFDE. For* $\gamma < 1$, *the comparison result is shown in Table 7, where both models maintain good agreement with the numerical solution, effectively capturing*

the nonlocal behavior of fractional systems, and the performance metrics result is shown in Table 8, showing that PINFDE consistently achieves lower error values compared to NFDE, highlighting its superior accuracy and robustness.

Table 5. The predicted values of proposed NFDE and proposed PINFDE at $\gamma = 1$.

t	Proposed NFDE	Proposed PINFDE
0	1	1
0.10	0.8186951875	0.8164060115
0.20	0.6705139875	0.6655565500
0.30	0.5496878623	0.5422190427
0.40	0.4512892365	0.4415790438
0.50	0.3712033629	0.3594347238
0.60	0.3060487508	0.2922590374
0.70	0.2530730366	0.2371634840
0.80	0.2100449204	0.1918142437
0.90	0.1751558184	0.1543344259
1.00	0.1469345688	0.1232109665

Table 6. Performance analysis of the models at $\gamma = 1$.

Method	R^2	NSE	MAE	MSE	SSE
Proposed NFDE	0.9989	0.9989	7.12×10^{-3}	6.78×10^{-5}	6.85×10^{-3}
Proposed PINFDE	0.9995	0.9995	4.55×10^{-3}	2.72×10^{-5}	2.75×10^{-3}

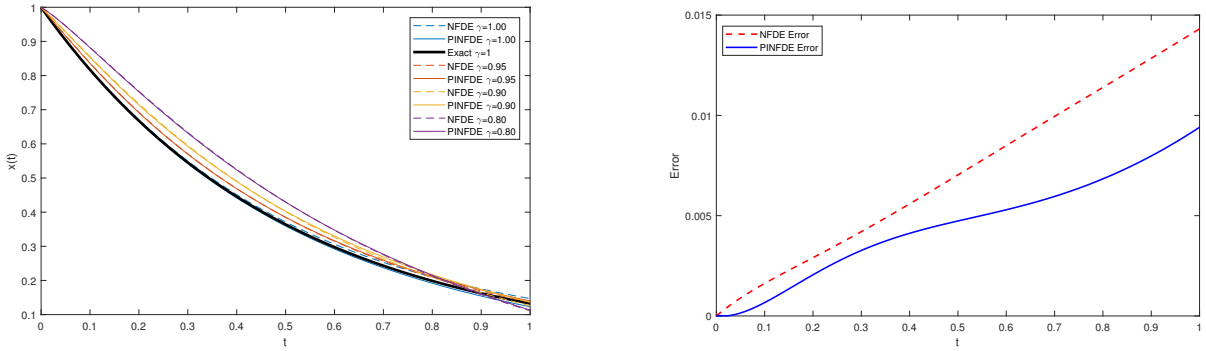
Table 7. Comparison table of numerical, NFDE, and PINFDE at $\gamma < 1$.

	$\gamma = 0.95$			$\gamma = 0.90$			$\gamma = 0.80$			$\gamma = 0.70$			$\gamma = 0.60$			$\gamma = 0.50$		
t	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE
0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	0.9920	0.9875	0.9916	0.9891	0.9852	0.9886	0.9861	0.9822	0.9849	0.9831	0.9791	0.9818	0.9800	0.9759	0.9787	0.9768	0.9728	0.9755
0.2	0.9801	0.9722	0.9793	0.9754	0.9686	0.9745	0.9701	0.9632	0.9683	0.9651	0.958	0.9633	0.9601	0.9532	0.9583	0.9551	0.9482	0.9533
0.3	0.9626	0.9531	0.9611	0.95715	0.94838	0.9556	0.9517	0.943	0.9495	0.9463	0.9376	0.9441	0.9410	0.9323	0.9388	0.9357	0.9270	0.9336
0.4	0.9404	0.9305	0.9385	0.9354	0.9255	0.9335	0.9304	0.9205	0.9285	0.9254	0.9155	0.9235	0.9204	0.9105	0.9158	0.9154	0.9055	0.9135
0.5	0.9120	0.8997	0.9099	0.9070	0.8945	0.9049	0.9020	0.8897	0.9000	0.8970	0.8847	0.8950	0.8920	0.8797	0.8900	0.8870	0.8747	0.8850
0.6	0.8806	0.8657	0.8788	0.8756	0.8607	0.8738	0.8706	0.8557	0.8688	0.8656	0.8507	0.8638	0.8606	0.8457	0.8588	0.8556	0.8407	0.8538
0.7	0.8456	0.8289	0.8442	0.8406	0.8245	0.8386	0.8356	0.8181	0.8336	0.8306	0.8131	0.8286	0.8256	0.8081	0.8236	0.8206	0.8031	0.8186
0.8	0.8109	0.7928	0.8097	0.8059	0.7888	0.8037	0.8009	0.7808	0.7987	0.7959	0.7758	0.7937	0.7909	0.7708	0.7887	0.7859	0.7658	0.7837
0.9	0.7750	0.7558	0.7737	0.7700	0.7516	0.7680	0.7650	0.7425	0.7630	0.7600	0.7375	0.7580	0.7550	0.7325	0.7530	0.7500	0.7275	0.7480
1.0	0.7403	0.7202	0.7386	0.7353	0.7152	0.7336	0.7303	0.7052	0.7286	0.7253	0.7002	0.7236	0.7203	0.6952	0.7186	0.7153	0.6902	0.7136

Table 8. Performance analysis of the models at $\gamma < 1$.

	NFDE					PINFDE				
γ	RMSE	MAE	SSE	R^2	NSE	RMSE	MAE	SSE	R^2	NSE
0.95	1.3062e-02	1.1834e-02	1.2113e-03	0.9670	0.9670	1.3553e-03	1.2333e-03	1.2440e-05	0.9997	0.9997
0.90	1.2331e-02	1.2150e-02	1.1063e-03	0.9799	0.9799	1.1468e-03	1.3150e-03	1.2170e-05	0.9997	0.9997
0.80	1.5271e-02	1.2823e-02	1.3299e-03	0.9741	0.9741	1.2183e-03	1.1627e-03	2.4050e-05	0.999	0.999
0.70	1.1427e-02	1.1283e-02	1.2399e-03	0.9752	0.9752	1.5483e-03	1.2567e-03	2.1524e-05	0.9996	0.9996
0.60	1.2527e-02	1.1283e-02	1.3879e-03	0.9752	0.9752	1.1283e-03	1.2567e-03	2.1240e-05	0.9996	0.9996
0.50	1.2751e-02	1.2133e-02	1.3894e-03	0.9757	0.9757	1.8134e-03	1.6721e-03	2.0457e-05	0.9997	0.9997

Figure 7a shows the predicted solutions at different values of γ, where it can be observed that the solution profiles vary smoothly with decreasing fractional order, reflecting the influence of memory effects in the system. NFDE and PINFDE models successfully capture their behavior, indicating their capability to approximate fractional dynamics accurately, and Figure 7b at $\gamma = 1$ shows the error analysis where the error remains low across the domain, confirming the accuracy and stability of the proposed method.



(a) Predicted solutions of the model at different values of γ **(b)** Error analysis at $\gamma = 1$

Figure 7. Predicted solutions and error analysis for the NFDE and PINFDE models.

Moreover, the PINFDE model exhibits lower error compared to the NFDE model, demonstrating the advantage of incorporating P-inf. constraints. Figure 8 shows the comparative analysis of our model with the numerical method at $\gamma < 1$, where both models closely follow the numerical solution across all fractional orders.

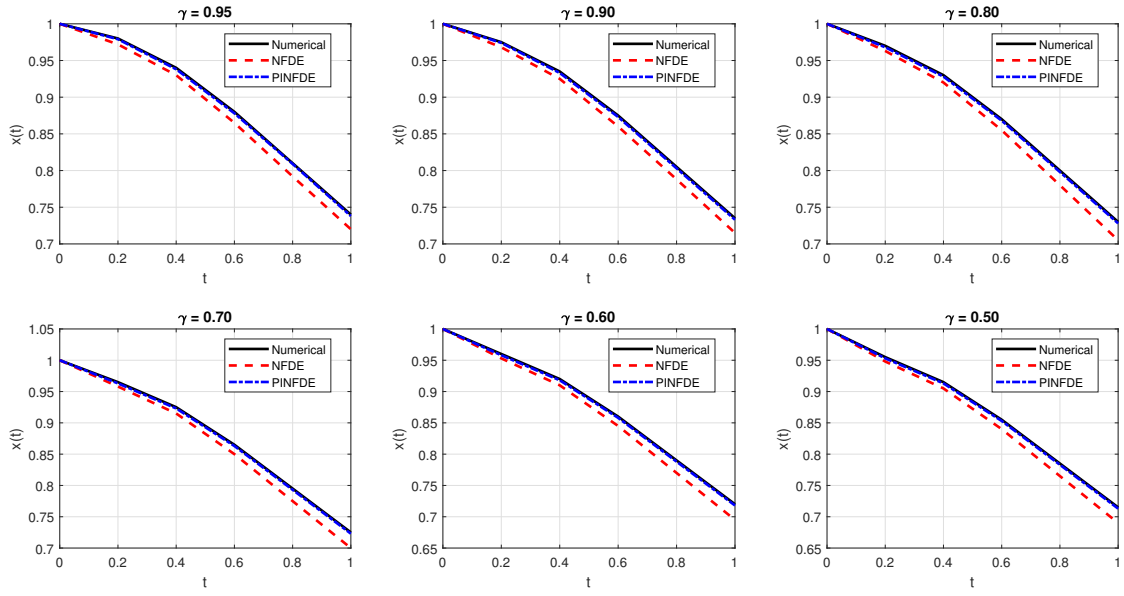


Figure 8. Comparison graph of the proposed NFDE and proposed PINFDE with the numerical solution at different values of $\gamma < 1$.

Particularly, PINFDE shows better results with the numerical solution, highlighting its improved accuracy and robustness in handling fractional-order systems and Figure 9 at $\gamma < 1$ presents the error analysis, confirming that the proposed NFDE accumulates higher errors, whereas the proposed PINFDE achieves smoother and lower error levels. Altogether, the findings establish that proposed PINFDE outperforms proposed NFDE by offering enhanced accuracy, stability, and reliability for modeling fractional-order systems.

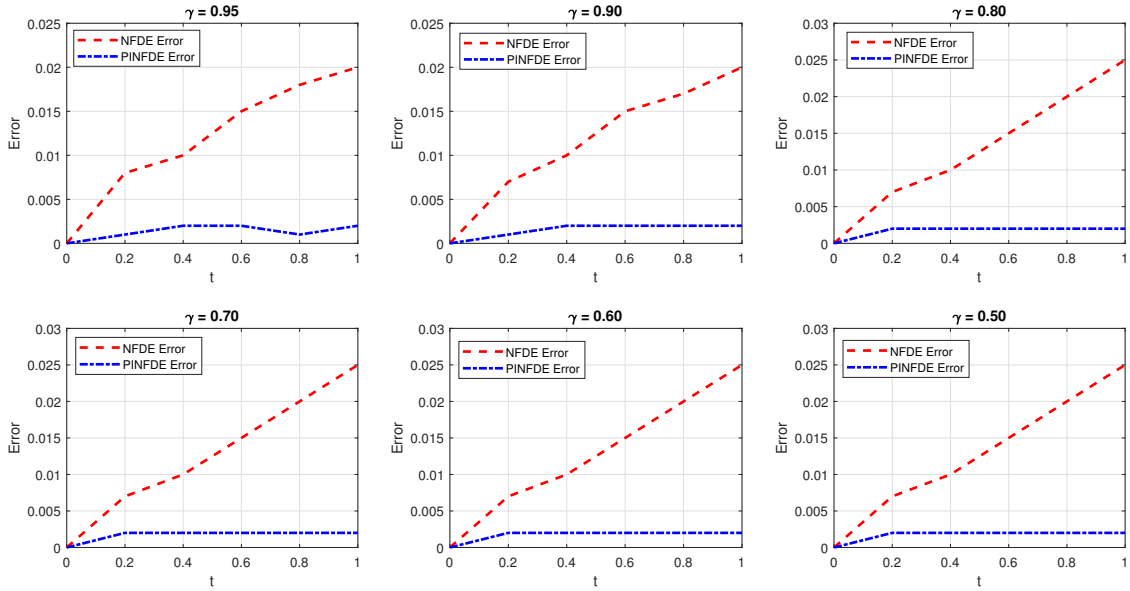


Figure 9. Error analysis at $\gamma < 1$.

Example 5.3. *Consider the fractional differential equation:*

$$^{CK}D^{\gamma,\beta}x(t) = -t^2x^2 + \sqrt{t} - x - e^x + 1,$$

where $x(0) = 1$ and $t \in [0, 1]$.

This example is solved using proposed NFDE and proposed PINFDE at $\gamma \in (0, 1]$. An accurate approximation of the solution is achieved by partitioning the interval $[0, 1]$ into 100 points. For the proposed PINFDE model, the physics term is taken as $-x$ for implementing the well-structured system, and the other term is used for the proposed NFDE model while adhering to the constraints imposed by the known physical term and the NN's training attempts for approximating the solution. To assess the efficacy of the proposed NFDE and proposed PINFDE models, different values of $\gamma = 0.95, 0.9, 0.8, 0.7, 0.6, 0.5$ are used. Table 9 shows the predicted value at $\gamma = 1$, where NFDE and PINFDE models produce accurate approximations of the solution. This is further supported by the performance metrics in Table 10, which indicate low error values and stable behavior of the models. Notably, the PINFDE model demonstrates improved accuracy compared to NFDE, owing to the incorporation of P-inf. constraints. Table 11 provides a comparison between numerical solutions and the predictions obtained using NFDE and PINFDE for different values of $\gamma < 1$, where both models are in good agreement with the numerical results, effectively capturing the nonlocal characteristics of fractional systems. Table 12 presents the corresponding performance analysis, where

PINFDE consistently yields lower error values than NFDE, highlighting its superior performance and robustness.

Table 9. The predicted values of proposed NFDE and proposed PINFDE term at $\gamma = 1$.

t	Proposed NFDE	Proposed PINFDE
0	1	1
0.10	0.7889257073	0.7876162528
0.20	0.6459338665	0.6455074548
0.30	0.5492958426	0.5497320294
0.40	0.4845730066	0.4853500127
0.50	0.4420083761	0.4426513910
0.60	0.4148567914	0.4150925874
0.70	0.3983439803	0.3981289267
0.80	0.3890246152	0.3885272145
0.90	0.3843805193	0.3839437961
1.00	0.3825614452	0.3826568722

Table 10. Performance analysis of the models at $\gamma = 1$.

Method	R^2	NSE	MAE	MSE	SSE
Proposed NFDE	0.9993	0.9993	1.07×10^{-2}	2.55×10^{-4}	1.78×10^{-2}
Proposed PINFDE	0.9996	0.9996	6.18×10^{-3}	1.06×10^{-4}	7.48×10^{-3}

Table 11. Comparison table of numerical, NFDE, and PINFDE at $\gamma < 1$.

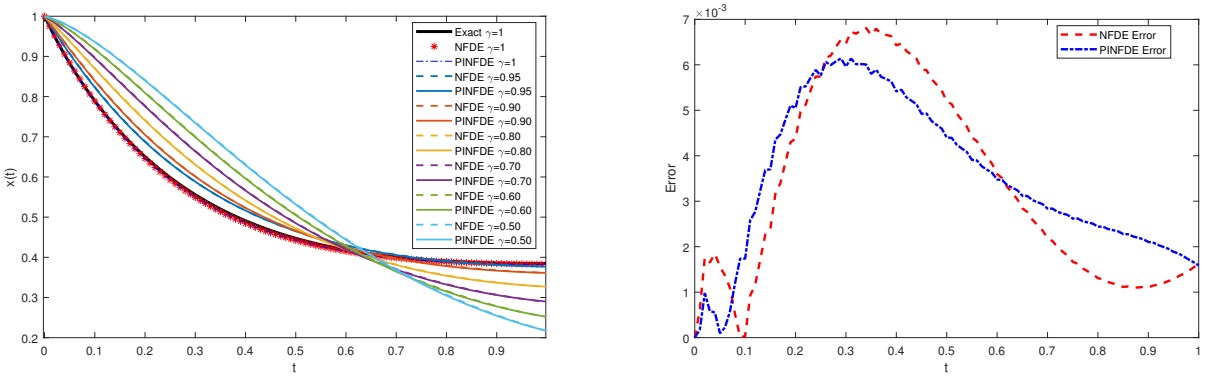
	$\gamma = 0.95$			$\gamma = 0.90$			$\gamma = 0.80$			$\gamma = 0.70$			$\gamma = 0.60$			$\gamma = 0.50$		
t	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE
0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	0.7508	0.7713	0.7516	0.7314	0.7492	0.7321	0.6897	0.7215	0.6908	0.6512	0.6796	0.6509	0.6205	0.6518	0.6213	0.5996	0.6317	0.6009
0.2	0.6093	0.6418	0.6114	0.5896	0.6212	0.5930	0.5708	0.6014	0.5722	0.5491	0.5797	0.5515	0.5407	0.5713	0.5421	0.5294	0.5596	0.5308
0.3	0.5211	0.5589	0.5224	0.5092	0.5487	0.5116	0.4995	0.5382	0.5013	0.5007	0.5415	0.5026	0.5004	0.5318	0.5017	0.5002	0.5326	0.5019
0.4	0.4597	0.5016	0.4618	0.4609	0.4898	0.4632	0.4694	0.5013	0.4716	0.4708	0.5099	0.4725	0.4803	0.5107	0.4820	0.4906	0.5124	0.4913
0.5	0.4305	0.4712	0.4317	0.4312	0.4696	0.4325	0.4509	0.4798	0.4523	0.4601	0.4894	0.4619	0.4706	0.5015	0.4724	0.4808	0.5022	0.4819
0.6	0.4103	0.4516	0.4118	0.4197	0.4512	0.4214	0.4395	0.4717	0.4416	0.4509	0.4818	0.4523	0.4702	0.4916	0.4715	0.4811	0.5014	0.4826
0.7	0.4006	0.4319	0.4014	0.4102	0.4311	0.4117	0.4298	0.4516	0.4315	0.4503	0.4707	0.4518	0.4609	0.4813	0.4616	0.4797	0.4908	0.4812
0.8	0.3904	0.4213	0.3921	0.3996	0.4217	0.4019	0.4302	0.4511	0.4318	0.4406	0.4609	0.4417	0.4605	0.4714	0.4613	0.4703	0.4912	0.4718
0.9	0.3912	0.4221	0.3928	0.4007	0.4225	0.4024	0.4309	0.4520	0.4323	0.4411	0.4615	0.4426	0.4610	0.4722	0.4619	0.4712	0.4920	0.4725
1.0	0.3908	0.4109	0.3917	0.4003	0.4116	0.4012	0.4306	0.4414	0.4309	0.4402	0.4507	0.4415	0.4601	0.4706	0.4608	0.4704	0.4811	0.4716

Table 12. Performance analysis of the models at $\gamma < 1$.

	NFDE					PINFDE				
γ	RMSE	MAE	SSE	R^2	NSE	RMSE	MAE	SSE	R^2	NSE
0.95	3.1464e-02	2.9150e-02	9.9260e-03	0.9726	0.9726	1.6432e-03	1.5025e-03	2.713e-05	0.9999	0.9999
0.90	2.6833e-02	2.4135e-02	7.2132e-03	0.9788	0.9788	1.5492e-03	1.4102e-03	2.4212e-05	0.9999	0.9999
0.80	2.6458e-02	2.4101e-02	7.0121e-03	0.9763	0.9763	1.6125e-03	1.4125e-03	2.6325e-05	0.9999	0.9999
0.70	2.7749e-02	2.5213e-02	7.7245e-03	0.9718	0.9718	1.5492e-03	1.4154e-03	2.4125e-05	0.9999	0.9999
0.60	2.3452e-02	2.1054e-02	5.5012e-03	0.9782	0.9782	8.9443e-04	8.1245e-04	8.0123e-06	1.0000	1.0000
0.50	2.1213e-02	1.9012e-02	4.5120e-03	0.9812	0.9812	9.4868e-04	9.0120e-04	9.2100e-06	1.0000	1.0000

Figure 10a presents the predicted solutions at different values of γ, where it can be observed that the

solution behavior varies smoothly as the fractional order decreases, reflecting the influence of memory effects inherent in fractional systems. NFDE and PINFDE models successfully capture this variation, demonstrating their capability to approximate generalized fractional dynamics. Figure 10b shows the error analysis at $\gamma = 1$, which shows that the error values remain relatively small, indicating the accuracy and stability of the proposed model. Additionally, the PINFDE model exhibits lower error compared to the NFDE model, highlighting the benefit of incorporating P-inf. constraints.



(a) Predicted solutions of the model at different values of γ **(b)** Error analysis at $\gamma = 1$

Figure 10. Predicted solutions and error analysis for the NFDE and PINFDE models.

Figure 11 presents the comparison between numerical solutions and the predicted solutions obtained using NFDE and PINFDE for different values of $\gamma < 1$. It can be observed that both models follow the numerical solution closely across all fractional orders, with PINFDE showing better alignment and improved accuracy.

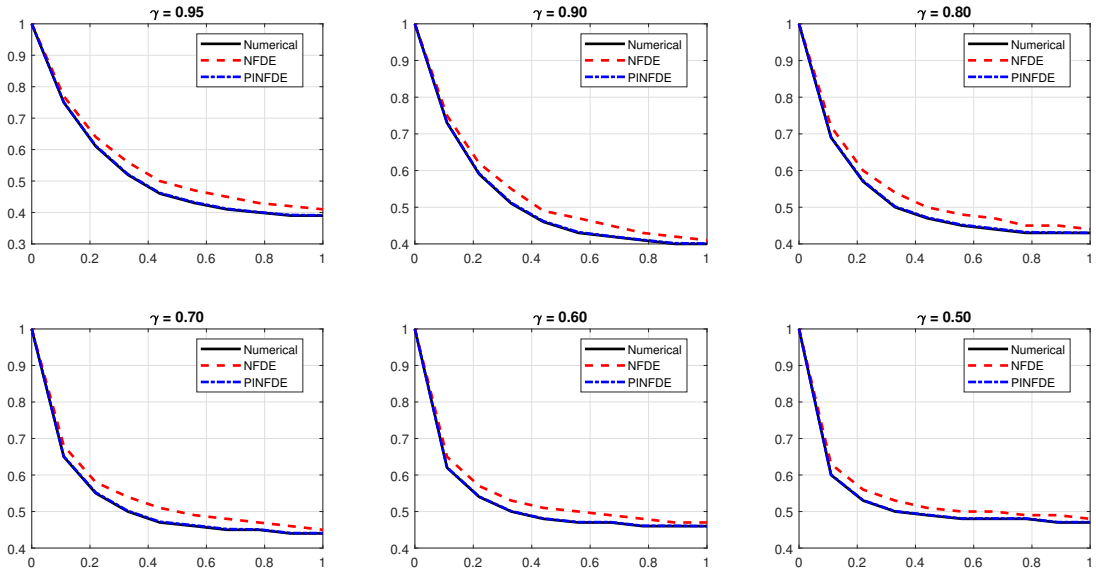


Figure 11. Comparison graph of the proposed NFDE and proposed PINFDE with the numerical solution at different values of $\gamma < 1$.

Figure 12 shows the corresponding error analysis at $\gamma < 1$, which shows that the error remains low and well-controlled across the domain, confirming the robustness and stability of the proposed approach. Moreover, the PINFDE model consistently achieves lower error compared to NFDE, demonstrating its superior performance in solving fractional-order systems.

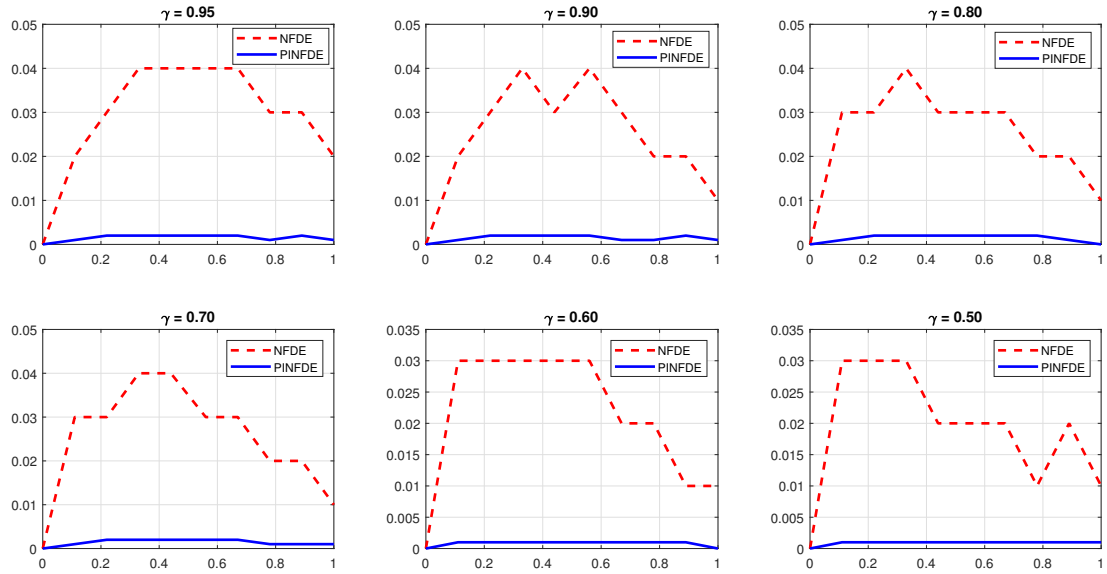


Figure 12. Error analysis at $\gamma < 1$.

Example 5.4. *To illustrate the proposed method, let us examine the fractional differential equation:*

$$^{CK}D^{\gamma,\beta}x(t) = t^2 - x - \cos(t) - \sin(t) + x^2 + \sin(2t) - \tan(x),$$

where $x(0) = 1$ and $t \in [0, 1]$.

The example is solved using the proposed NFDE and proposed PINFDE at $0 < \gamma \leq 1$. For the proposed PINFDE model, the physics term is taken as $-x$ for implementing the well-structured system, and the other term is used to predict the proposed NFDE model. NN training seeks to get the most accurate solution while adhering to the constraints imposed by a given physical term. Table 13 presents the predicted values at $\gamma = 1$, where the NFDE and PINFDE models yield accurate approximations of the solution. This is also supported by the performance analysis shown in Table 14, which shows low error values, confirming the reliability and stability of the proposed model. The PINFDE model demonstrates improved accuracy compared to NFDE due to the incorporation of P-inf. constraints. To assess their behavior, the proposed NFDE and proposed PINFDE models are tested at different γ values, i.e., $\gamma = 0.95, 0.9, 0.8, 0.7, 0.6, 0.5$, and are compared with the numerical method shown in Table 15. The results indicate that both models effectively capture the fractional dynamics, maintaining better agreement with the numerical solution. Table 16 shows the corresponding performance metrics, where the PINFDE model consistently achieves lower error values than NFDE, highlighting its superior accuracy and robustness.

Table 13. The predicted values of proposed NFDE and the proposed PINFDE at $\gamma = 1$.

t	Proposed NFDE	Proposed PINFDE
0	1	1
0.10	0.7688244581	0.7716404795
0.20	0.5796728134	0.5843009948
0.30	0.4293022155	0.4351201057
0.40	0.3139301538	0.3206378817
0.50	0.2296016216	0.2370836138
0.60	0.1724531650	0.1806411743
0.70	0.1388743519	0.1476484537
0.80	0.1255902051	0.1347231864
0.90	0.1296902298	0.1388282775
1.00	0.1486243605	0.1572924852

Table 14. Performance analysis of the models at $\gamma = 1$.

Method	R^2	NSE	MAE	MSE	SSE
Proposed NFDE	0.9993	0.9993	1.62×10^{-2}	3.67×10^{-4}	2.57×10^{-2}
Proposed PINFDE	0.9998	0.9998	7.96×10^{-3}	9.72×10^{-5}	6.80×10^{-3}

Table 15. Comparison table of numerical, NFDE, and PINFDE at $\gamma < 1$.

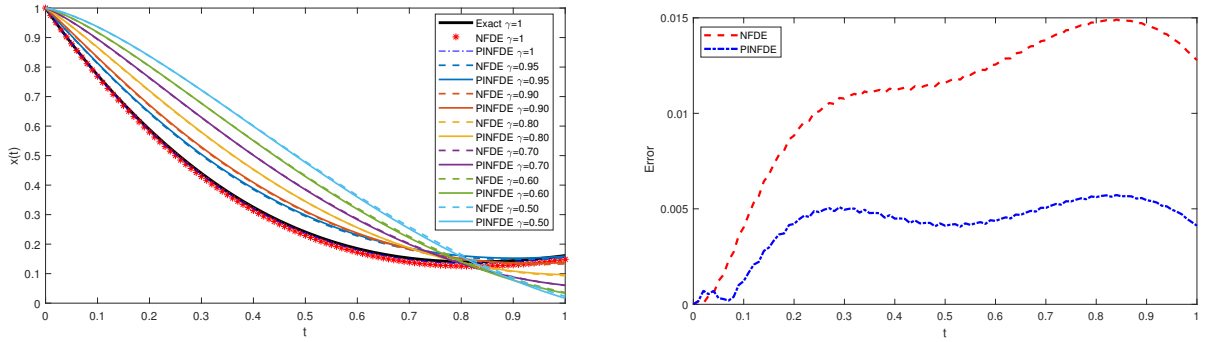
	$\gamma = 0.95$			$\gamma = 0.90$			$\gamma = 0.80$			$\gamma = 0.70$			$\gamma = 0.60$			$\gamma = 0.50$		
t	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE	Numerical	NFDE	PINFDE
0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	0.7012	0.7387	0.7134	0.6823	0.7117	0.6914	0.6345	0.6745	0.6467	0.5767	0.6116	0.5898	0.5397	0.5726	0.5447	0.4912	0.5323	0.5056
0.2	0.5212	0.5645	0.5387	0.5021	0.5443	0.5145	0.4422	0.4823	0.4567	0.4045	0.4473	0.4198	0.3665	0.4076	0.3712	0.3589	0.3998	0.3622
0.3	0.3678	0.4022	0.3712	0.3445	0.3887	0.3511	0.3021	0.3423	0.3134	0.2856	0.3256	0.2967	0.2689	0.3009	0.2789	0.2617	0.3056	0.2715
0.4	0.2698	0.3045	0.2712	0.2554	0.2987	0.2643	0.2312	0.2789	0.2456	0.2294	0.2687	0.2378	0.2367	0.2776	0.2498	0.2423	0.2843	0.2512
0.5	0.1923	0.2398	0.2023	0.1932	0.2378	0.2056	0.1912	0.2345	0.2078	0.2016	0.2422	0.2198	0.2218	0.2698	0.2376	0.2488	0.2890	0.2523
0.6	0.1612	0.2045	0.1723	0.1645	0.2012	0.1745	0.1867	0.2244	0.1967	0.2089	0.2409	0.2187	0.2397	0.2734	0.2498	0.2645	0.3011	0.2722
0.7	0.1476	0.1867	0.1534	0.1534	0.1834	0.1676	0.1912	0.2267	0.2056	0.2234	0.2523	0.2387	0.2587	0.2934	0.2698	0.2909	0.3308	0.3034
0.8	0.1545	0.1912	0.1623	0.1776	0.1988	0.1767	0.2134	0.2412	0.2287	0.2512	0.2834	0.2698	0.2987	0.3276	0.3023	0.3398	0.3623	0.3412
0.9	0.1611	0.1923	0.1609	0.1723	0.1965	0.1778	0.2145	0.2487	0.2245	0.2556	0.2865	0.2624	0.2987	0.3223	0.3023	0.3385	0.3609	0.3454
1.0	0.1723	0.2145	0.1812	0.2045	0.2223	0.2123	0.2465	0.2734	0.2567	0.2987	0.3278	0.3056	0.3398	0.3656	0.3434	0.3709	0.4034	0.3867

Table 16. Performance analysis of the models at $\gamma < 1$.

	NFDE					PINFDE				
γ	RMSE	MAE	SSE	R^2	NSE	RMSE	MAE	SSE	R^2	NSE
0.95	1.3720e-03	3.5000e-02	1.3654e-02	0.9819	0.9819	9.8975e-05	9.9921e-03	9.2314e-04	0.9988	0.9988
0.90	1.0600e-03	3.3812e-02	1.0600e-02	0.9852	0.9852	8.2654e-05	8.1210e-03	8.0829e-04	0.9989	0.9989
0.80	1.2300e-03	3.3000e-02	1.2300e-02	0.9807	0.9807	9.1275e-05	9.2658e-03	9.3857e-04	0.9986	0.9986
0.70	1.2345e-03	3.3845e-02	1.3150e-02	0.9784	0.9784	9.9125e-05	9.6543e-03	9.1235e-04	0.9984	0.9984
0.60	1.3125e-03	3.4152e-02	1.3840e-02	0.9749	0.9749	9.0278e-05	9.0354e-03	9.0221e-04	0.9983	0.9983
0.50	1.3256e-03	3.4121e-02	1.3254e-02	0.9727	0.9727	9.1284e-05	9.8654e-03	9.1275e-04	0.9981	0.9981

Figure 13a compares the performance of the proposed NFDE model and the proposed PINFDE model for different fractional orders γ ranging from 1.0 *to* 0.5. *The reference solution (exact at $\gamma = 1$) is shown in solid black, while proposed NFDE results are indicated with markers and solid lines,*

and proposed PINFDE results are shown with dashed lines. From the results, it is observed that the proposed NFDE and proposed PINFDE are capable of capturing the solution's overall decay behavior. At $\gamma = 1$, the proposed NFDE matches closely with the exact solution, validating the model accuracy in the integer-order case. However, when γ decreases ($\gamma = 0.95, 0.9, 0.8, \cdots, 0.5$), deviations become more apparent. In general, the proposed PINFDE yields smoother and more stable approximations than the proposed NFDE, especially when the fractional orders are far from unity. Overall, this figure shows that the proposed NFDE can approximate the solution well in the conditions it has seen during training, but the proposed PINFDE is more robust and consistent over different fractional orders, demonstrating an evident advantage by encoding physical information into the learning process. Figure 13b plots the error comparison of the proposed NFDE and PINFDE at $\gamma = 1$. The red dashed curve is the proposed NFDE error, while the proposed PINFDE error is the blue dashed curve.



(a) Predicted solutions of the model at different values of γ **(b)** Error analysis at $\gamma = 1$

Figure 13. Predicted solutions and error analysis for the NFDE and PINFDE models.

It can be observed that the proposed NFDE accumulates more error as time increases, while the proposed PINFDE remains approximately very small over the time interval. This indicates that the proposed PINFDE continuously yields more accurate predictions and reduces error propagation in time compared to the proposed NFDE, revealing the advantage of incorporating physical constraints in the NN model. Figure 14 depicts the comparison results of the proposed NFDE and PINFDE with the numerical solution at $\gamma < 1$. The error analysis at $\gamma < 1$ is shown in Figure 15, which shows that the PINFDE model consistently exhibits lower error compared to the NFDE model throughout the domain, demonstrating its improved accuracy and effectiveness due to the incorporation of the P-inf. constraints. The result also shows that the error behavior remains well-controlled as the fractional order decreases, confirming the robustness of the proposed method for solving fractional-order systems.

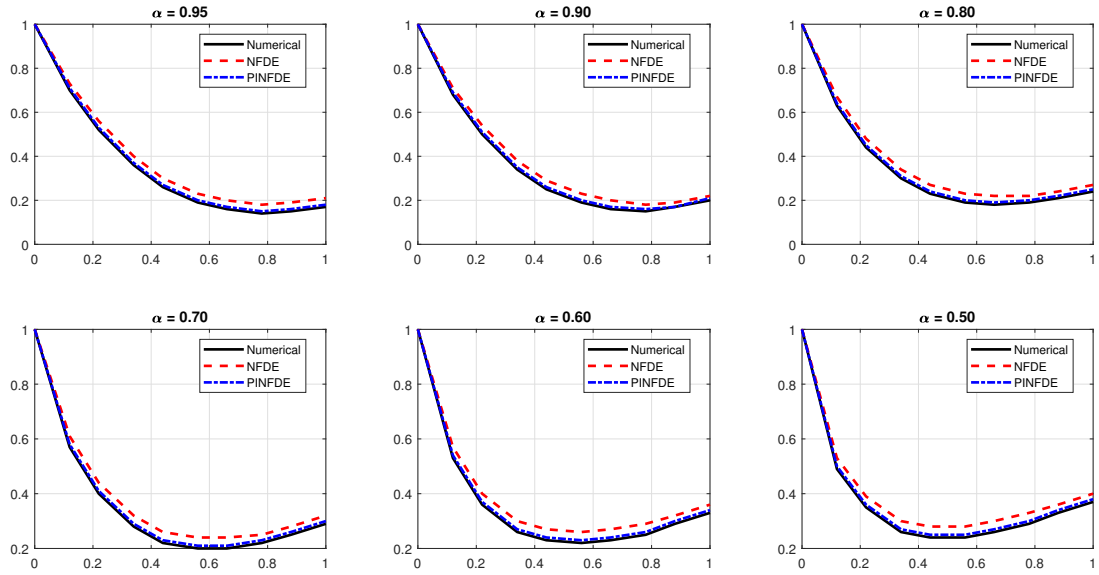


Figure 14. Comparison graph of the proposed NFDE and proposed PINFDE with the numerical solution at different values of $\gamma < 1$.

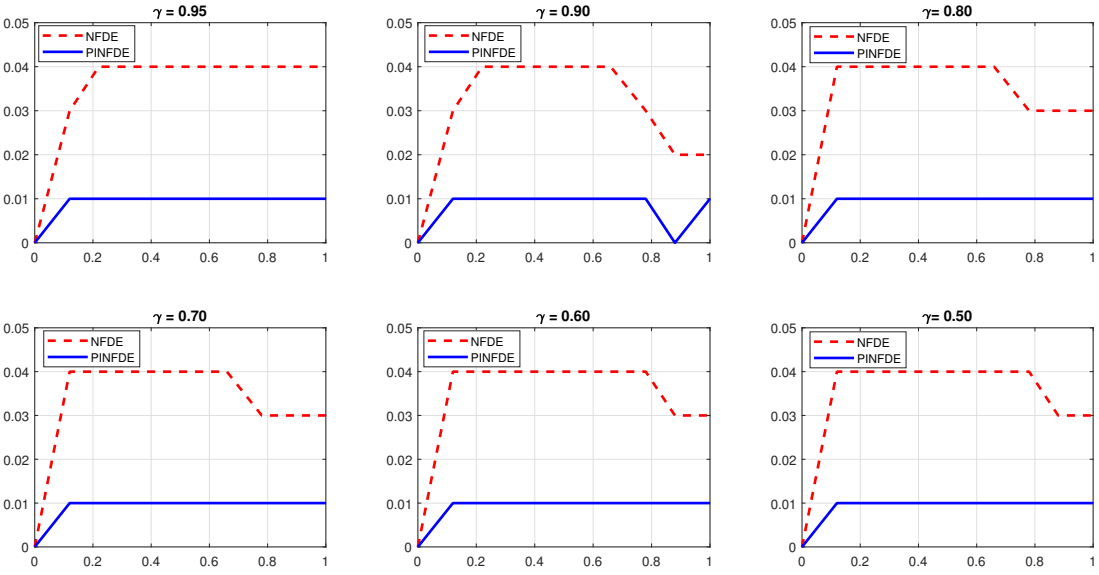


Figure 15. Error analysis at $\gamma < 1$.

6. Discussion and conclusions

6.1. Discussion

In this paper, we have investigated the solution of FDEs governed by the C–K fractional derivative using both NFDEs and their P-inf. extension (PINFDEs). The incorporation of fractional calculus into the neural framework enables the capture of memory effects and hereditary properties that are essential for accurately modeling complex systems. To further enhance the performance of the NNs, we adopted the SiLU A/F, which was observed to improve accuracy, stability, and convergence compared to traditional A/F, thereby demonstrating the flexibility of neural architectures in handling fractional-order dynamics.

In this work, we utilized the Volterra integral equation of the C–K derivative. This enabled the incorporation of memory effects in a natural way and enabled the use of quadrature-based discretization for improved numerical stability. Training of the models was performed with the Adam optimization algorithm, which efficiently enabled weight and bias updates, thus guaranteeing robust convergence. The synergy between this integral formulation and the fractional derivatives, combined with advanced optimization, enabled the proposed framework to reach a high level of accuracy in approximating the solutions. Extensive numerical simulations were performed to demonstrate the applicability and efficiency of the proposed NFDE and PINFDE models. In Example 5.1, a comparison was made with the exact solution for $\gamma = 1$ and with the numerical solution for $\gamma < 1$, showcasing that our approach achieves accurate results. Moreover, the evaluation of performance metrics and analysis of error graphs showed that the errors obtained from the proposed models are considerably fewer, thus justifying the effectiveness of the framework. The results underline that the proposed PINFDEs are able to outperform purely data-driven proposed NFDEs by embedding P-inf. constraints.

Despite these advantages, certain limitations remain. The computational cost associated with training NNs for fractional systems can be significant, particularly for complex or high-dimensional problems. Moreover, the performance of the proposed framework depends on the selection of network architecture, A/F, and hyperparameters. Additionally, this study is primarily validated for benchmark examples, and its effectiveness for large-scale or real-world applications requires further investigation.

6.2. Conclusions and future work

In this work, we have shown that the combination of the Caputo-Katugampola derivative with the NFDE and PINFDE architectures, driven by SiLU A/F and modern optimization methods, will provide an effective, precise, and flexible paradigm for fractional differential equations. The approach does not only contribute to error reduction and increasing accuracy but also shows the wide range of possibilities offered by neural fractional models in modeling discrepancies for scientific and engineering applications.

In future research, we will focus on extending the proposed framework for NFDE and PINFDE to higher dimensions and multi-term FDEs. Additionally, different training strategies and architectures will be explored for better accuracy and efficiency. Further investigation into hybrid approaches that combine NNs with traditional numerical methods, as well as applications to real-world problems in engineering and applied sciences, represents promising directions for future work.

Author contributions

Sneha Agarwal: Conceptualization, methodology, validation, formal analysis, investigation, resources, writing—original draft preparation, writing—review and editing, visualization; Lakshmi Narayan Mishra: Conceptualization, methodology, validation, formal analysis, investigation, resources, writing—original draft preparation, writing—review and editing, visualization, supervision. All authors contributed equally to this work and have carefully reviewed the manuscript.

Use of Generative-AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare that they have no competing interest.

References

1. Z. Yao, Z. Yang, J. Gao, Unconditional stability analysis of Grünwald Letnikov method for fractional-order delay differential equations, *Chaos Soliton Fract.*, **177** (2023), 114193. https://doi.org/10.1016/j.chaos.2023.114193
2. P. Řehák, On decaying and asymptotically constant solutions of nonlinear equations with the Weyl fractional derivative of an order in (1, 2), *Appl. Math. Lett.*, **145** (2023), 108779. https://doi.org/10.1016/j.aml.2023.108779
3. M. Ansari, L. N. Mishra, Analysis of Ulam-Hyers stability and the existence of solutions in nonlinear Caputo fractional differential equations involving integral boundary conditions, *Korean J. Math.*, **34** (2026), 21–40. https://doi.org/10.11568/kjm.2026.34.1.21
4. V. K. Pathak, L. N. Mishra, V. N. Mishra, On the solvability of a class of nonlinear functional integral equations involving Erdélyi–Kober fractional operator, *Math. Methods Appl. Sci.*, **46** (2023), 14340–14352. https://doi.org/10.1002/mma.9322
5. Y. Chen, B. Hosseini, H. Owhadi, A. M. Stuart, Solving and learning nonlinear PDEs with Gaussian processes, *J. Comput. Phys.*, **447** (2021), 110668. https://doi.org/10.1016/j.jcp.2021.110668
6. A. Antony, L. N. Mishra, A novel neural network-based method for solving fractional differential equations involving the generalized Caputo operator, *Adv. Stud.: Euro-Tbil. Math. J.*, **19** (2026), 233–252. https://doi.org/10.32513/asetmj/193220082619115
7. Y. Li, T. Wang, G. H. Gao, The asymptotic solutions of two-term linear fractional differential equations via Laplace transform, *Math. Comput. Simul.*, **211** (2023), 394–412. https://doi.org/10.1016/j.matcom.2023.04.010
8. M. T. Hanna, Discrete fractional Hankel transform based on a nonsymmetric kernel matrix, *Digit. Signal Process.*, **126** (2022), 103431. https://doi.org/10.1016/j.dsp.2022.103431

9. I. A. Bhat, L. N. Mishra, V. N. Mishra, Comparative analysis of nonlinear Urysohn functional integral equations via Nyström method, *Appl. Math. Comput.*, **494** (2025), 129287. https://doi.org/10.1016/j.amc.2025.129287
10. C. Nwaigwe, A. Atangana, Construction and analysis of numerical algorithms for Erdelyi-Kober fractional integral equations, *Sci. Afr.*, **28** (2025), e02756. https://doi.org/10.1016/j.sciaf.2025.e02756
11. M. Ruggieri, M. P. Speciale, Asymptotic expansion method with respect to a small parameter for fractional differential equations with Riemann-Liouville derivate, *Commun. Nonlinear Sci. Numer. Simul.*, **138** (2024), 108234. https://doi.org/10.1016/j.cnsns.2024.108234
12. V. M. José, R. Luis, B. G. Gerardo, Optimal control of linear fractional differential equations in the Caputo sense, *J. Comput. Appl. Math.*, **470** (2025), 116707. https://doi.org/10.1016/j.cam.2025.116707
13. T. He, T. Zhai, X. Huang, M. Li, The parareal algorithm based on a new local time-integrator for nonlinear Caputo–Hadamard fractional differential equations, *Commun. Nonlinear Sci. Numer. Simul.*, **152** (2026), 109183. https://doi.org/10.1016/j.cnsns.2025.109183
14. I. A. Bhat, L. N. Mishra, A comparative study of discretization techniques for augmented Urysohn type nonlinear functional Volterra integral equations and their convergence analysis, *Appl. Math. Comput.*, **470** (2024), 128555. https://doi.org/10.1016/j.amc.2024.128555
15. C. Guler, A new numerical algorithm for the Abel equation of the second kind, *Int. J. Comput. Math.*, **84** (2007), 109–119. https://doi.org/10.1080/00207160601176889
16. M. Vellappandi, S. Lee, Physics-informed neural fractional differential equations, *Appl. Math. Model.*, **145** (2025), 116127. https://doi.org/10.1016/j.apm.2025.116127
17. A. Antony, L. N. Mishra, The role of residual neural networks for advancing fractional differential equations, *Bol. Soc. Parana. Mat.*, **43** (2025), 1–24. https://doi.org/10.5269/bspm.76732
18. R. Ali, Z. Zhang, H. Ahmad, Exploring soliton solutions in nonlinear spatiotemporal fractional quantum mechanics equations: an analytical study, *Opt. Quantum Electron.*, **56** (2024), 838. https://doi.org/10.1007/s11082-024-06370-2
19. H. Ren, X. Meng, R. Liu, J. Hou, Y. Yu, A class of improved fractional physics informed neural networks, *Neurocomputing*, **562** (2023), 126890. https://doi.org/10.1016/j.neucom.2023.126890
20. B. Wang, J. Sun, B. Peng, X. Cui, L. Cheng, X. Zheng, Optimal event-triggered neural learning tracking control for pneumatic muscle antagonistic joint with asymmetric constraints, *IEEE Trans. Ind. Electron.*, **72** (2025), 14677–14687. https://doi.org/10.1109/TIE.2025.3585055
21. F. Jarad, T. Abdeljawad, D. Baleanu, On the generalized fractional derivatives and their Caputo modification, *J. Nonlinear Sci. Appl.*, **10** (2017), 2607–2619. https://doi.org/10.22436/jnsa.010.05.27
22. S. Agarwal, L. N. Mishra, Attributes of residual neural networks for modeling fractional differential equations, *Heliyon*, **10** (2024), e38332. https://doi.org/10.1016/j.heliyon.2024.e38332
23. L. Sadek, S. A. ldris, F. Jarad, The general Caputo–Katugampola fractional derivative and numerical approach for solving the fractional differential equations, *Alex. Eng. J.*, **121** (2025), 539–557. https://doi.org/10.1016/j.aej.2025.02.065

24. W. Mei, X. Wang, Y. Lu, K. Yu, S. Li, Learning and current prediction of PMSM drive via differential neural networks, *IEEE Trans. Circuits Syst. II: Express Br.*, **72** (2025), 489–493. https://doi.org/10.1109/TCSII.2025.3527024

25. N. Zhao, T. Zhao, J. Cao, H. Shi, Fault diagnosis of dual-network recursive interval type-2 fuzzy neural network based on SVD-TLS optimization, *Comput. Ind. Eng.*, **207** (2025), 111280. https://doi.org/10.1016/j.cie.2025.111280

26. I. F. Merizio, F. R. Chavarette, T. C. Moro, R. Outa, V. N. Mishra, Machine learning applied in the detection of faults in pipes by acoustic means, *J. Inst. Eng. (India): C*, **102** (2021), 975–980. https://doi.org/10.1007/s40032-021-00682-y

27. X. Xu, B. Li, PDE-based observation and predictor-based control for linear systems with distributed infinite input and output delays, *Automatica*, **170** (2024), 111845. https://doi.org/10.1016/j.automatica.2024.111845

28. Z. Fu, W. Xu, S. Liu, Physics-informed kernel function neural networks for solving partial differential equations, *Neural Netw.*, **172** (2024), 106098. https://doi.org/10.1016/j.neunet.2024.106098

29. Z. Hu, K. Kawaguchi, Z. Zhang, G. E. Karniadakis, Tackling the curse of dimensionality in fractional and tempered fractional PDEs with physics-informed neural networks, *Comput. Methods Appl. Mech. Eng.*, **432** (2024), 117448. https://doi.org/10.1016/j.cma.2024.117448

30. H. D. Mazraeh, K. Parand, GEPINN: An innovative hybrid method for a symbolic solution to the Lane-Emden type equation based on grammatical evolution and physics-informed neural networks, *Astron. Comput.*, **48** (2024), 100846. https://doi.org/10.1016/j.ascom.2024.100846

31. M. Sratia, A. Oulmelk, L. Afraites, A. Hadri, M. A. Zaky, A. S. Hendy, On a computational paradigm for a class of fractional order direct and inverse problems in terms of physics-informed neural networks with the attention mechanism, *J. Comput. Sci.*, **85** (2025), 102514. https://doi.org/10.1016/j.jocs.2024.102514

32. C. Coelho, M. F. P. Costa, L. L. Ferrás, Neural fractional differential equations, *Appl. Math. Model.*, **144** (2025), 116060. https://doi.org/10.1016/j.apm.2025.116060

33. A. Shafiq, A. B. Colak, T. N. Sindhu, Optimization of micro-rotation effect on magnetohydrodynamic nanofluid flow with artificial neural network, *Z. Angew. Math. Mech.*, **104** (2024), e202300498. https://doi.org/10.1002/zamm.202300498

34. W. Sawangtong, P. Dunnimit, B. Wiwatanapataphee, P. Sawangtong, An analytical solution to the time fractional Navier-Stokes equation based on the Katugampola derivative in Caputo sense by the generalized Shehu residual power series approach, *Partial Differ. Equ. Appl. Math.*, **11** (2024), 100890. https://doi.org/10.1016/j.padiff.2024.100890

35. A. L. Martire, Volterra integral equations: An approach based on Lipschitz-continuity, *Appl. Math. Comput.*, **435** (2022), 127496. https://doi.org/10.1016/j.amc.2022.127496